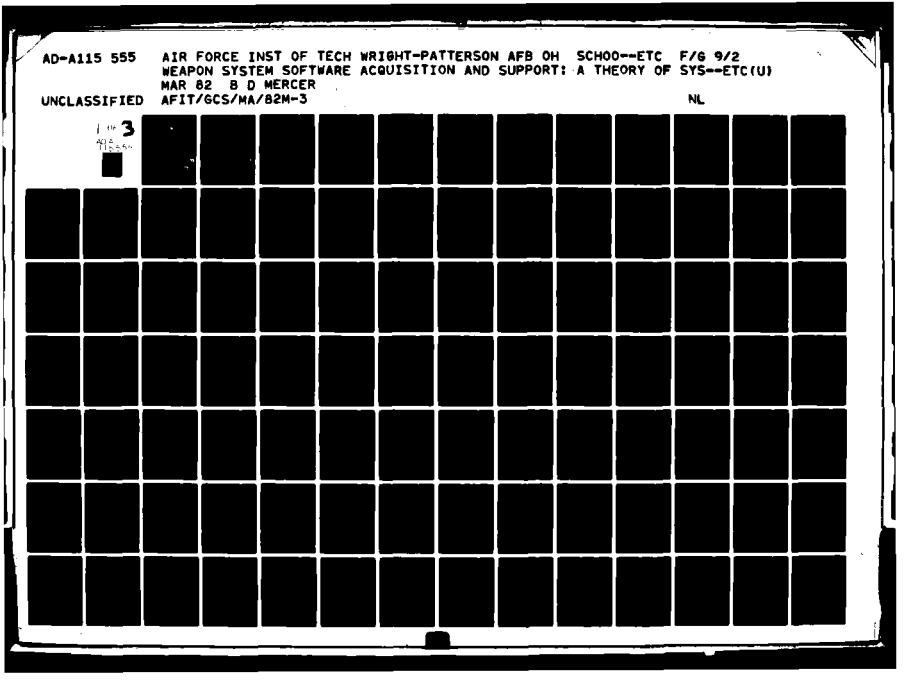
AD-A115 555 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYS--ETC(U)
MAR 82 B D MERCER

UNCLASSIFIED AFIT/GCS/MA/82M-3 NL



DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/MA/82M-3

(1)

WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYSTEM STRUCTURE AND BEHAVIOR

THESIS

AFIT/GCS/MA/82M-3 Bradford D. Mercer
Capt USAF

Approved for public release; distribution unlimited

AFIT/GCS/MA/82M-3

WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYSTEM STRUCTURE AND BEHAVIOR

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Bradford D. Mercer, B.A., M.S.S.M
Capt USAF

Graduate Computer Systems

March 1982

Preface

The purpose of this study was to provide a vehicle in the form of a model or theory of structure to aid policy makers at all levels in increasing their understanding of the complexities and interactions underlying the decisional structure and information network of the weapon system software acquisition and support system. Nowhere has this researcher found an application of the dynamic feedback structural analysis of system dynamics applied to the arena of software management and engineering.

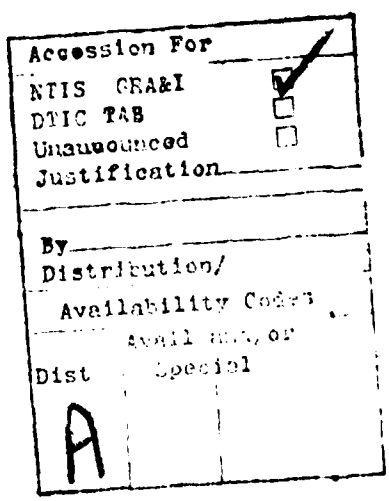
Although conceptual in nature, I believe this work to be foundational to the development of a validated dynamic approach to aiding policy makers in the management of software development and support endeavors.

It is with greatest sincerity that I extend my thanks to Professor Daniel E. Reynolds, Assistant Professor of Mathematics and advisor to this research, and Lieutenant Colonel Thomas D. Clark, Jr., Associate Professor of Systems Management, both of the Air Force Institute of Technology, for their guidance, understanding and support in the completion of this study. The insight and enlightenment provided by these two gentlemen, along with their continuing dialogue concerning the nature of such research,

has provided me with invaluable tools for continued research and success in many future endeavors, as well as lifelong friendships.

Finally, I must recognize my wife, Donna, for her very special support, inspiration and caring during this most intense effort. Her sacrifice was tremendous, for which I will always be grateful.

— Bradford D. Mercer



Contents

		Page
Preface		ii
List of Figures		vii
List of Tables		x
Abstract		xi
I.	Introduction	1
	Weapon System Embedded Computers	1
	The Significance of Software	3
	Making the Right Choices	7
II.	Problem Description and Methodology	9
	Problem Description	9
	Research Methodology	11
	A Note on Experimental and Nonexperimental Research	14
III.	System Dynamics' Foundations and Techniques	17
	The System Dynamics Philosophy	17
	Previous Research Related to this Study	20
	The System Dynamics Focus on Policy	21
	Systems Dynamics as a Theory of Structure in Systems	22
	Causal Loop Diagramming	25
	Feedback Loop Behavior	29
	System Dynamics Flow Diagramming	35
	Equation Writing	35
	Chapter Summation	41
IV.	The Traditional View of the Weapon System Software Acquisition and Support Process	42
	Introduction	42
	Acquisition of Major Defense Systems	42
	Conceptual Phase	44
	Validation Phase	45

Page

Full-Scale Development Phase 45
Deployment Phase 47

Less Elaborate Acquisitions 48
The Computer Program Life Cycle 49
Chapter Summation 49

V. A View of the Weapon System Software Acquisition and Support Process Based Upon a Theory of its Structure 52

Definition of a Context for Discussing Embedded Computer Systems 52
Measurement of System Effectiveness 61
Development of a Conceptual Structure 67
A System Dynamics Flow Diagram of the Generalized Software Production Loop 74

VI. Conceptual Influences Upon the Structure and Behavior of the Weapon System Software Production Process 78

Introduction 78
Overall Monitor and Control of the Process--The Production Management Sector . 78
Primary Influences on the Software Production Rate--The Technology Sector and the Manpower and Productivity Sector . . 85

The Technology Sector 85
The Manpower and Productivity Sector 92

Primary Influences on the Error and Rework Discovery Rates--The Quality Management Sector 100
Chapter Summation 104

VII. A Mathematical Model of the Weapon System Software Acquisition and Support System . . . 106

Introduction 106
Production Sector 107
Production Management Sector 111
Technology Sector 120
Manpower and Productivity Sector 127
Quality Management Sector 143
Chapter Summation 149

Page

VIII. Computerization, Verification and Experimentation 150

Computerization 151
An Approach to Verification and Validation 152
A View Towards Experimentation 157

IX. Conclusions and Recommendations 159

Bibliography 166

Appendix A: DYNAMO Model Equations 170

Vita 180

List of Figures

Figure		Page
1-1.	Trend in the Use of Airborne Digital Computers	2
1-2.	Hardware/Software Cost Trend	5
2-1.	Inquiry in System Dynamics	16
3-1.	Information Conversion and Decision Modulation by Policy	22
3-2.	Causal Loop Diagram of Decision Making	26
3-3.	Simple Feedback System	27
3-4.	Directional Linkages	28
3-5.	Polarity of Feedback Loops	28
3-6.	Positive Feedback Behavior	30
3-7.	Negative Feedback Behavior, Adjustment Without Oscillation	32
3-8.	More Complex Feedback Loop, Exhibiting Damped Oscillatory Behavior	33
3-9.	Stable and Explosive Oscillatory Patterns of Negative Feedback Behavior	34
3-10.	System Dynamics Flow Diagram Symbols	36
3-11.	Causal Loop Diagram and Translated Flow Diagram	38
4-1.	System Life Cycles	50
5-1.	Contextual System Environment for Embedded Computer Acquisition and Support	53
5-2.	A Conceptual View of the ECS Acquisition and Support Process and its Environment	57
5-3.	Causal Loop Diagram of Weapon System Software Acquisition and Support	60

Figure		Page
5-4.	Weapon System Software Acquisition and Support within the Context of Software Engineering and Management	66
5-5.	Simplified Software Production Process	69
5-6.	The Generalized Software Production Loop . . .	70
5-7.	Cost and Difficulty in Correction of Errors .	73
5-8.	System Dynamics Flow Diagram of the Generalized Software Production Loop	75
5-9.	An Expanded System Dynamics Model of the Generalized Software Production Process . . .	77
6-1.	Control System Structure of Organizations . .	80
6-2.	Production Management Sector	82
6-3.	Growth in Integrated Circuit Technology . . .	86
6-4.	Management Capability to Implement Technology	88
6-5.	Technological and Management Growth in Generations	89
6-6.	Technology Sector	91
6-7.	Manpower and Productivity Sector	94
6-8.	Time Versus Number of Workers--Partitionable Task Requiring Communication	98
6-9.	Time Versus Number of Workers--Task with Complex Interrelationships	99
6-10.	Quality Management Sector.	103
7-1.	Production Sector	108
7-2.	Production Management Sector	112
7-3.	Table of Work Required Estimation Error versus Management Effectiveness	115
7-4.	Table of Production Pressure versus a Comparison of Overrun to Schedule	117

Figure		Page
7-5.	Table of Pressure to Relax Requirements or Schedule versus Production Pressure	119
7-6.	Technology Sector	121
7-7.	Table of Pressure for Improved Technology versus a Comparison of a Perceived Gap to the State of the Art	124
7-8.	Manpower and Productivity Sector (Part 1). . .	128
7-9.	Manpower and Productivity Sector (Part 2). . .	136
7-10.	Table of Proportion of Engineers in Testing versus Quality Improvement and Production Pressures	140
7-11.	Table of Man-Month Modifiers versus Total Production Engineers and Intrinsic Effort Required	143
7-12.	Quality Management Sector	145
7-13.	Table of Quality Improvement Pressure versus Excess Error Rate	148
9-1.	A Conceptual View of Policy Oriented Decision Support	164

List of Tables

Table		Page
1.	Software Quality Attributes	101
2.	Production Sector Variables	109
3.	Production Management Sector Variables	113
4.	Technology Sector Variables	122
5.	Manpower and Productivity Sector Variables (Part 1)	129
6.	Manpower and Productivity Sector Variables (Part 2)	137
7.	Quality Management Sector Variables	146

Abstract

The system for acquiring and supporting weapon system software was investigated through the methodology of system dynamics, a technique for studying the structure of socio-technical systems and how that structure determines their behavior. Conceptual and mathematical models of a generalized software production process and the influences upon that process were developed. The mathematical model was translated into a continuous simulation computer model using the DYNAMO language.

These models can be examined by managers at all levels in the acquisition and support process in order to increase their understanding of the complexities and interactions of system decisional structures. Such understanding is foundational to the analysis and formulation of policy to guide and improve the performance of that management system.

Discussion of experimental and nonexperimental modes of research lead to recommendations for future experimental research for policy analysis, formulation, and implementation.

WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYSTEM STRUCTURE AND BEHAVIOR

I. Introduction

Weapon System Embedded Computers

Digital electronics are used in virtually every Air Force weapon system. Such electronics form the *brain*, *sensors*, *effectors*, and the connecting *nervous system* of these weapon systems. The key component in such installations is the digital computer. The use of digital computers in aircraft systems began in the late 1950s with the installation of the MA-1 fire control system on the F-106 interceptor aircraft. The MA-1, considered advanced for the time, used electronic tube technology coupled with a rotating drum memory (Ref 40:22). Since this primitive beginning, the influence and complexity of airborne digital computers has been ever increasing; so that today such technology plays a dominant role in aircraft development, use and capability. This increasing trend is shown in Figure 1-1.

Several factors have produced this trend, but it is primarily due to the rapidly increasing productivity in digital technologies, which has permitted the cost of

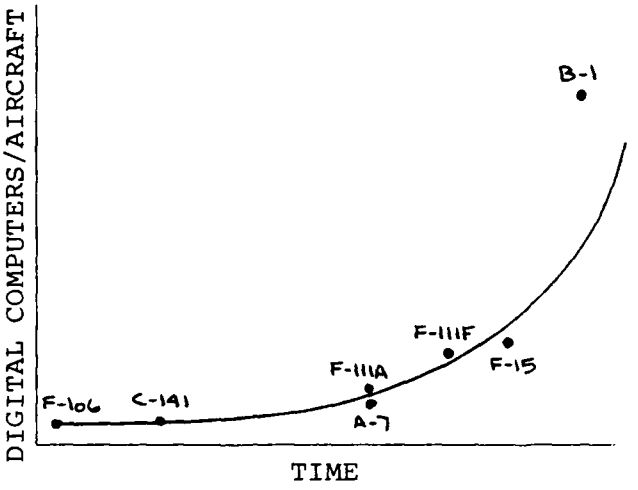


Fig. 1-1. Trend in the Use of Airborne Digital Computers [Ref 39:227]

digital implementations to drop sharply as compared with other possible mechanizations. The spectrum and flexibility of applications for digital computers have also influenced the trend. Computers are not only being used in large, centralized functions of avionics (e.g., navigation and fire control), but are being employed in a distributed fashion throughout these aircraft to implement many smaller subsystem functions (e.g., engine air inlet control) (Ref 39:227).

Today the weapon system digital computer is referred to as an embedded computer system (ECS) because it is incorporated as an integral part of the weapon system's identity and function.[1] The embedded computer

[1]The full definition of embedded computer reads, "A computer incorporated as an integral part of, dedicated to, or required for direct support of, or for the upgrading or modification of, major or less than major systems." In

has made it possible to increase the capability and effectiveness of aircraft weapon systems, while maintaining the requirements for operation and control of those systems within the capability of their human operators (i.e., the aircrew).

Within a general trend, this increase in capability and effectiveness has made it possible for a single weapon system to perform the functions assigned to multiple weapon systems of previous generations, and to reduce the number of individual weapon system units required to accomplish a particular mission. From this we may assume that embedded computer systems serve to multiply the apparent size and effectiveness of our air force, while maintaining or reducing the apparent system complexity presented to the users of that force.[2]

The Significance of Software

The expansion of applications for programmable embedded computers in airborne weapon systems has created a corresponding increase in requirements for weapon system computer software. This software performs functions of

---

reference to aircraft systems this definition also includes any automatic test equipment (ATE), support systems for software maintenance, and aircrew training devices (ATD) (Ref 15:15).

[2]In the sense of this statement, embedded computers are a manifestation of the application of Ashby's Cybernetic Law of Requisite Variety. An excellent discussion of this concept is contained in Chapter 12--"Coping with Complexity," of Stafford Beer's Decision and Control (Ref 4).

overriding importance. First, software integrates weapon system and embedded computer into a functional system with high reliance on digital sensing and control. The software in controlling the operation of the embedded computer system essentially controls the capability of the weapon system. Errors in software design or miscalculation of onboard data are eventually reflected in weapon system control and utilization. Embedded computer software capability directly relates with mission accomplishment and weapon system effectiveness (Ref 8:143).

There is increasing interest in weapon system software on the part of government activities, such as the Government Accounting Office and the Office of Management and Budget. This is especially true of the Department of Defense. This interest is due in part to the extremely high cost of weapon system software. In fact, the monies and effort dedicated to the acquisition and support of embedded computer software is tremendously greater than that for the associated hardware.

This trend of increasing cost for software was predicted by Boehm (Ref 5) in 1973, and is illustrated in Figure 1-2.

In the Air Force, weapon system software was initially ignored.

> If it was considered at all, it was assumed to be an item of no real importance. It was thought of as an element which automatically arrived in the Air Force inventory with hardware avionics [Ref 22:662].

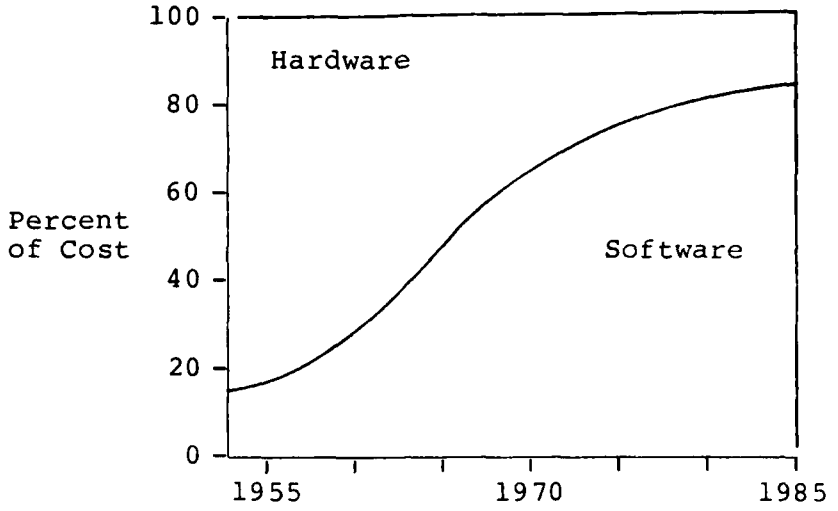


Fig. 1-2. Hardware/Software Cost Trend

During the 1970s the Air Force became aware that the cost of software was an increasing proportion of the total cost of airborne electronics.

> Toward the end of the 1970's, it was estimated that Air Force expenditure for avionics software was between $1 billion and $1.5 billion, while the expenditure for hardware was about $300 million or $400 million [Ref 22:662].

The cost of airborne software was exceeding three times the cost of hardware.

Within the Department of Defense, Jacques Gansler, former Deputy Assistant Secretary of Defense for Material Acquisition, stated in April 1976 that a Pentagon study

> . . . identified 115 different defense systems that employ "embedded computers" of which approximately one-half are now in service and the other half are under development [Ref 37:43].

Gansler continues by saying, "According to our estimate, the Pentagon is spending more than $3 billion annually for software for defense systems [Ref 37:41]." He went on to

state that 68 percent of the amount is spent during system development and 32 percent is spent for operating and maintenance costs. Another estimate of DOD software costs, six months earlier in October 1975, stated that, "Current annual expenditures for embedded computer systems exceed $2 billion, with more than 70 percent of this amount dedicated to software [Ref 16]."

This tremendous cost for embedded computer software is only one aspect of its significance. A second important aspect is unsatisfactory software performance. A well-known expert in the field Mr. Barry C. DeRoze, of the Directorate of Weapons Support Systems Acquisition, Office of the Assistant Secretary of Defense, Installations and Logistics, has said,

> Although hardware reliability has improved substantially, the corresponding gains in system reliability have not been realized. This apparent contradiction arises because software unreliability--the failure of software to satisfy the stated operational requirements--has been the "tall-pole in the tent" in determining the reliability and operational readiness of systems [Ref 16:3].

In an attempt to place the dual problems of software cost and performance in perspective, the Department of Defense and the Air Force have directed increased emphasis on the total cost of an embedded computer system over its life cycle (Refs 10-14). Many studies have been done which show that quality and reliability are major factors in the life cycle cost of software systems. Dr. David S. Alberts, in a MITRE Corporation report, states that,

conservatively, 50 percent of total life cycle cost of software systems is attributable to defects. Moreover, of the error types studied, he reports, "Design errors were found to contribute just over 80 percent of the total cost of error [Ref 2]."

Making the Right Choices

Why has the cost and effort required for the acquisition and support of computer software come to dominate the design, development and use of weapon system embedded computer systems? The rapid increases in productivity in digital hardware technologies, which permitted the cost of digital implementations to follow a trend towards decreasing costs have been overcome by a low or reversing rate of productivity in software technology. The development of software is labor-intensive, which works against the economies of scale required to implement ever increasing complexity and size in embedded computer applications.

A continuous theme in discussing weapon system acquisition and support problems has been a perception of inadequate management or a lack of proper management emphasis on the software area. This view is substantiated by General Robert T. Marsh, Commander of the Air Force Systems Command (AFSC), when, as commander of AFSC's Electronic Systems Division, he stated, "In many ways, our problem has shifted from one of technology itself to one

of how to manage the technology and how to make the right choices [Ref 26:7]."

"Making the right choices" implies having a strategy for making choices. Strategy formulation is a function of higher levels of management. Strategy results from policy making, which outlines the decision rules for managing. Thus, I may now present the motivating thesis of this paper--that the solutions to the problems of weapon systems software acquisition and support will result from the analysis and formulation of policy to guide and improve the performance of that management system.

II. Problem Description and Methodology

Benjamin Franklin's reply to a lady who queried the usefulness of his work on electricity: "Madam, what use is a newborn baby?"

— *Arthur Koestler: The Ghost in the Machine*

Problem Description

How can it be that the problems apparent in the acquisition and support of weapon systems software continue to persist, and that the magnitude of the cost and performance failures in such projects continues to increase?

Don't managers at all levels influencing software acquisition and support formulate and implement policies designed to guide these systems towards the goal of quality, affordable weapon systems software?

The problems of cost and performance have been studied extensively by direct investigation. The examination of such problems and the determination of solution techniques constitutes the major effort in the disciplines of software management and engineering. Perhaps the answers to the above questions can be obtained by considering the way in which such investigations are conducted. Underlying the apparent difficulties may be a general lack of understanding of the complexities and interactions

in the entire process of weapon systems software acquisition and support.

Some researchers have emphasized pieces rather than the process of software development and support. Others have emphasized techniques for estimating cost and effort rather than understanding of the complexities and interactions underlying the determination of cost, effort and performance. Many times such techniques involve only special formats for organizing accounting information. That information cannot be meaningfully utilized in an environment that lacks knowledge of the basic mechanisms producing such information.

Discussion and research into the framework of software development and support, by dividing such efforts into phases of work, has overemphasized the discrete nature of that work. Indeed such project life cycles can be viewed, at least after the fact, as having been composed of such segments. However, the dynamic essence, the behavior over time, of the process is distorted. The emphasis is upon discrete sets of activities separated in time and lacking any base of underlying common elements to bind them. From this it is clear, that the fundamental systems nature of the process is ignored. The ever-present and controlling feedback between action, results, information, and new action is overlooked by such an approach.

A need exists for an alternative way of thinking about, studying and managing software acquisition and support: an approach based upon the systems nature of the process, acknowledging the process flow and feedback controlling aspect of such projects. Such an approach is outlined in the methodology following and detailed in succeeding chapters.

Research Methodology

The general methodology of this research is based upon the study of the structure of socioeconomic systems and how structure determines their behavior. This field of study is known as system dynamics. System dynamics began as industrial dynamics, which was described by its developer, Dr. Jay W. Forrester, as:

> . . . [treating] the time-varying (dynamic) behavior of industrial organizations. . . . Industrial dynamics is the study of the information-feedback characteristics of industrial activity to show how organization structure, amplification (in policies), and time delays (in decision and actions) interact to influence the success of enterprise [Ref 18:13].

Systems dynamics is particularly suited for the study of complex systems problems, in which many factors are interrelated through organizational information feedback paths. It is similarly well-suited for dynamic problems, in which the problematic aspects of the process tend to evolve and reveal themselves over a period of time. The two characteristics of system complexity and

time-process orientation are clearly typical of software acquisition and support activities.

The stages through which a system dynamics study progresses form the methodology and outline of this study. The first stage is the recognition of a socio-technical problem that appears to merit dynamic system-oriented investigation. The foregoing discussion clearly establishes the acquisition and support of weapon system software as such a problem area.

The second stage of a system dynamics study consists of the development of a verbal theory of cause-and-effect interaction and a verbal description of the policies in effect in the system being studied. A diagrammatic technique known as causal loop analysis is used to provide a visual treatment of the material. Chapters V and VI of this study develop such a theory and description.

The third stage requires the construction of a mathematical model of the weapon system software acquisition and support process. The basic flows of resources and the integrating network of information flow apparent in socio-technical systems are presented in such a model. This mathematical model is introduced in Chapter V and developed in Chapter VII. Each section of Chapter VII develops in a flow diagram form a visual treatment of material presented in the chapter. A mathematical interpretation and discussion of model parameters follows each

diagram. This model is intended to add precision and verification to, and hence increased understanding of, the descriptive and theoretical portions of this study.

The fourth stage of a system dynamics study is the computerization of the mathematical model and its use to generate simulated system behavior over time. The computer simulation is implemented using the DYNAMO[3] simulation language. Chapter VIII describes this stage and some initial behavior explorations.

The next stages of a system dynamics study are repeatedly iterative. The results of an initial behavior exploration are used in redesigning system parameters or policies, and incorporating the same into the mathematical model, followed by computer runs to determine the effect on the behavioral outcome of software acquisition and support projects. Chapter VIII also addresses this stage.

Finally, the goal of a system dynamics study is to not only contribute to understanding, but also to the solution of a real-world problem. Such a study should attempt to operationalize its policy recommendations, indicated by model experimentation, to improve the

---

[3]The DYNAMO compiler is a translator to FORTRAN for compiling and running continuous simulation models. It was developed by the Industrial Dynamics Group at M.I.T. in the late 1950s for simulating dynamic feedback system models. Its maturation closely parallels that of the system dynamics discipline, so that today's version is extensively revised from the original [Ref 30].

organization supporting the system of interest. The general nature of this study of software acquisition and support, rather than implementing its results within a specific operating entity, attempts to make appropriate a discussion of broad policy implications for Department of Defense and Air Force software managers.

A Note on Experimental and Nonexperimental Research

System dynamics is based firmly in the tradition of the scientific method. Freely interpreted, the paradigm for scientific research comprises six or seven phases: observation, hypothesis formulation, the design of an experiment to test the hypothesis, comparison of experimental results with those predicted by the hypothesis, hypothesis refinement, and a return to the beginning of the sequence (Ref 3:92).

It is important to note that the system dynamics approach draws from both experimental and nonexperimental modes of research, and, as such, influences perceptions of model purpose and validation. In experimental research, a model is a means to an end and typically is used as a format to generate and test hypotheses through controlled experiment. In nonexperimental research, the model itself becomes the focus, serving as a theory of the description and operation of a system. The transition in the system dynamics approach from the third to the fourth

and later stages predicates a shift from a nonexperimental to an experimental mode, and a change in perception from conceptual to developmental or technical purpose (Ref 38: 48-49).

A visual review of inquiry in system dynamics and the range of perception and mode of research is presented in Figure 2-1.

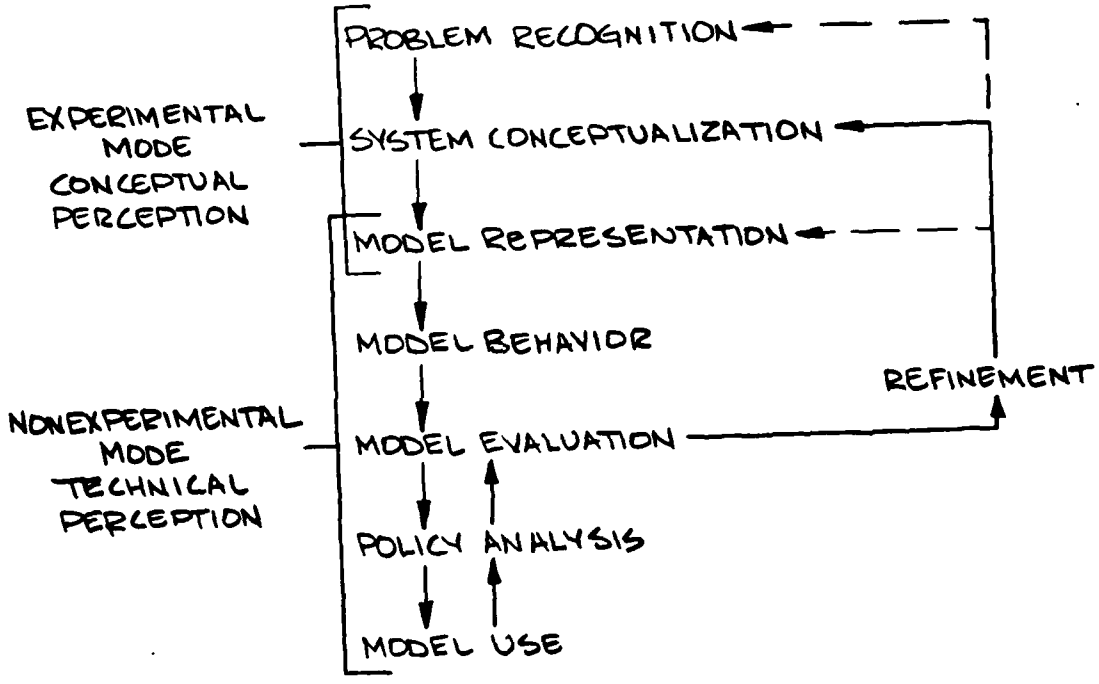


Fig. 2-1. Inquiry in System Dynamics (Ref 3:93)

III. System Dynamics' Foundations and Techniques

The System Dynamics Philosophy

System dynamics (originally industrial dynamics) was developed at the Massachusetts Institute of Technology (M.I.T.) during the 1950s, primarily by Jay W. Forrester. Concepts from four, then relatively new fields, were brought together to develop a guiding philosophy for the analysis and simulation of complex, nonlinear, multiloop feedback systems. These fields and their contributions were:

1. *Control systems engineering*--the concepts of feedback and system self-regulation;

2. Cybernetics--the nature of information and its role in control systems;

3. Organizational theory--the structure of human organizations and the form of human decision making; and

4. Digital simulation--the use of reliable, high speed digital computers to simulate large industrial-technical organizations.

The system dynamics philosophy is founded on the belief that the dynamic tendencies of an organization, that is, the behavioral patterns generated over time, are principally caused by the organization's structure.

Edward B. Roberts, one of Forrester's original industrial dynamics students, states that structure,

> . . . includes not only the physical aspects of plant and production process, but more importantly the policies and traditions, both tangible and intangible, that dominate decision making in that organization. Such a structural framework contains sources of amplification, time lags, and information feedback similar to those found in complex engineering systems. Engineering and management systems containing these characteristics display complicated response patterns to relatively simple system or input changes. The analysis of large nonlinear systems of this sort is a major challenge . . . effective and reliable redesign of such a system is still more difficult. The subtleties and complexities in the management area make these problems even more severe . . . the structural orientation of system dynamics provides a beginning for replacing confusion with order [Ref 33:4].

The second foundation of the system dynamics philosophy is that organizations are most effectively viewed in terms of their common underlying flows instead of separate functions. Forrester postulated that the flows of people, money, materials, orders and capital equipment, and the integrating flows of information can be identified in all organizations (Ref 18). Roberts continues by stating that,

> The flow structure orientation causes the viewer (manager or analyst) to cross suborganization boundaries in a natural manner. It acts to dispel the component approach to organization that promotes interorganizational conflict and unrecognized suboptimization. A meaningful system framework results from tracing cause-and-effect chains through the relevant flow paths [Ref 33:5].

A system dynamics methodology or paradigm was developed for practicing this evolving philosophy. Flow diagramming, mathematical modelling, and digital simulation

have been adopted and altered to fit the needs of practitioners. Cause-and-effect flow diagramming was developed from electrical engineering signal flow graphs to visually present the underlying structural situation.

Formalized flow diagramming and mathematical equation-writing were created for the next step, where such flow diagrams and equations represent organizational relationships as two categories--levels and rates. Levels represent those aspects of the real world in which accumulations of resources exist, such as inventories (of goods or ideas), balances of funds, pools of employees. Rates include all activities or controls of flow within the system, such as flows of effort, streams of information, payments for expenses, hiring of new employees.

Computer simulation of the levels and rates model of the organizational situation is employed next. DYNAMO is the predominantly used language for such simulations, although any general purpose language, such as FORTRAN can be used (but with great expense to programming efficiency). Simulation serves to experimentally determine ideas for policy improvement and test proposed policy changes.

As the system dynamics philosophy and the methods devised by Forrester and his system dynamics group developed at M.I.T., the results of that work began to diffuse to industry and management science at large. System

dynamics has since been applied to a wide variety of social, economic, and technical systems. The literature of system dynamics contains numerous descriptions of such models.[4]

Previous Research Related to this Study

Forrester originally applied system dynamics to problems of industrial management. The first system dynamics models addressed such general management problems as inventory fluctuations, instability of the labor force and falling market share (Ref 18). As soon as system dynamics work was initiated at M.I.T., practitioners began to see potential application in the analysis and improvement of technical organizations and research and development activities.

As an extension of Abraham Katz' M.I.T. master's thesis (Ref 23) and his own doctoral dissertation, Edward Roberts published the first comprehensive book describing a system dynamics study of research and development (Ref 32). Public awareness of other system dynamics models in the research and development area has been restricted, because most such models have been carried out in house without outside consulting advice. However, published reports in various annual editions of the MIT System

---

[4]The Most recent and complete bibliography of system dynamics literature is contained in Volume 14, of TIMS Studies in Management Science, Systems Dynamics (Ref 25). This source lists 41 books and 209 published articles at the time of its 1980 publication.

Dynamics Newsletter indicate that modelling efforts based upon the Robert's model (Ref 32) have been undertaken by SONY Labs, Raytheon, FMC Inorganic Chemicals Division, Motorola Military Electroncs, Hughes Aircraft, General Dynamics/Fort Worth, Grumman and IBM, among others.

Pugh-Roberts Associates, Inc., the major consulting firm in the field, have developed system dynamics models for several hundred industrial and governmental clients. That firm developed a huge model for Litton Industries of the entire design, development, and engineering phases of their naval shipbuilding programs (Ref 7). This model was intended to identify responsibility for the major cost overruns and schedule slippages in those programs.

The System Dynamics Focus on Policy

System dynamics focuses on policy, as well as structure, and how policy determines behavior. Forrester defined policy as a criteria for decision making, a rule that states how operating decisions are to be made (Ref 18:93).

Decisions made at any particular time result in action. Each decision, itself, is the result of applying policy to the particular conditions that prevail at the moment of decision. These conditions are transmitted to the decision maker as information. Over time and at a

high level of aggregation, decision making can be viewed as a stream or flow of decisions, made on the basis of a flow of information about particular conditions, and resulting in continual actions. Policy, then, is the rationale that determines how a stream of decision will be modulated in response to changing inputs of information. A feedback representation of this process is shown in Figure 3-1 (Ref 18:93-102).

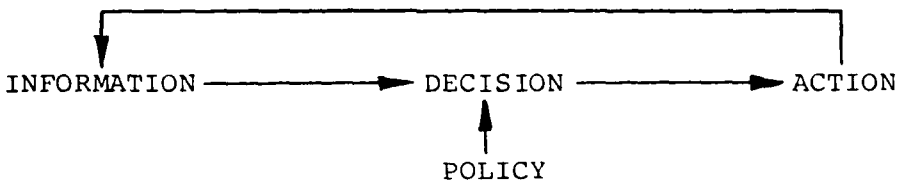


Fig. 3-1. Information Conversion and Decision Modulation by Policy

To describe policy means that one must describe the output and purpose of the decision stream that is controlled by that policy, and the inputs of information to the decision stream. In other words, one must describe the setting of interrelated policies and therefore the structure of a system. System dynamics, therefore, deals with policy, as well as structure, and with the resulting behavior (Ref 20:8).

Systems Dynamics as a Theory of Structure in Systems

The feedback loop apparent in Figure 3-1 of information-decision-action-new information forms the

fundamental construct through which one can, in viewing system dynamics as a theory of structure, systematically develop and analyze dynamic models of socio-technical systems. In Forrester's definition of system dynamics in Chapter II of this study, the feedback structure in conjunction with time delays was postulated as playing the key role in the description and response of a wide variety of systems. Forrester later noted,

> Feedback processes [have] emerged as universal in social systems and seemed to hold the key to structuring and clarifying relationships that had remained baffling and contradictory [Ref 19:399].

System dynamics views system structure as existing within four levels of a hierarchy. At the highest level, this theory deals with closed systems. The intended meaning is that the dynamic behavior of a system is generated within the boundaries of a defined system. This does not imply that nothing crosses the system boundary to affect the closed system, only that whatever crosses the boundary is not essential in creating the particular behavior being explored (Ref 19:406).

Within the system boundary, at the next lower level of the hierarchy, the system is viewed as composed of one or more feedback loops. The loops interact to produce the system behavior. The complexity of the system of interest is determined by the multiplicity and interrelationships of the feedback loops (Ref 19:406).

At the third level of the hierarchy, loops are composed of *level* and *rate* variables. In this view, levels define the state of the system at any moment, while rates are the flow or activity variables of the system. The rates are the policy statements in a system. Following Forrester's definition of policy, they are the rules whereby the state of the system (information) determines action (Ref 19:407).

At the lowest level of the hierarchy, a policy statement or rate is seen as describing four components. The first is the goal of the decision-making process, the objective towards which this part of the system, represented by a single feedback loop, is striving. The second, is the information input concerning the apparent state of the system on which the decision-making process is based. Third, the policy describes a process for determining a discrepancy between the goal and the apparent condition. Fourth, the policy describes an action which will result from the discrepancy (Ref 19:207).

The application of this theory of structure to the analysis and improvement of systems results in the system dynamics methodology described earlier in this chapter and in Chapter II. The determination of the system boundary is inclusive in the identification and definition of the problem to be considered. Defining a system boundary

focuses attention on what must be included to generate the dynamic behavior of a system.

While defining the boundary, the perception of the next level of structure dealing with feedback loops is initiated. This results in a verbal description and causal loop analysis of the system as the second stage of the methodology.

After the boundary and feedback loops are established, system variables are sorted into levels and rates. The result is a mathematical interpretation of the system. The addition of rate components as quantitative policy statements results in a parametric model of the system of interest, ready for computerization and policy experimentation.

The remainder of this chapter presents specific techniques for accomplishing causal loop diagramming and mathematical modelling of a system.

Causal Loop Diagramming

The feedback loops forming the central structures of a system dynamics model can be illustrated using a technique known as causal loop diagramming. Causal loop diagrams fulfill two roles in system dynamics modelling. First, during model conceptualization they serve as preliminary sketches of the causal hypotheses of the system structure producing dynamic behavior. Second, causal

loop diagrams, as a compact representation of system structure, simplify the illustration of a model. Figure 3-2 presents a simplified causal loop diagram of decision making.

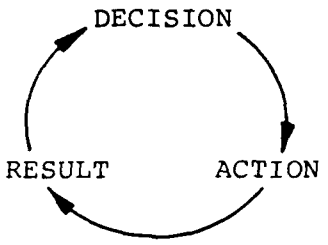


Fig. 3-2. Causal Loop Diagram of Decision Making

In Figure 3-2, decision, action, and result are considered as variables changing over time and connected by linkages reflecting the cause-and-effect relationship between two variables. It is important to note that causality is unidirectional from element to element around the loop. Two or more linkages connected in such a way that beginning with one variable a path can be traced following the arrows around and returning to the starting variable, as in Figure 3-2, constitute a feedback loop.

Implicit in every causal loop is a time delay: delay from each decision to its consequence as action, delay from each action to its consequence as a result, and delay in feeding back information about each result to affect the next decision.

A feedback system, Figure 3-3, consists of two or more connected causal or feedback loops. The behavior of the variables in each loop can propagate through the interconnections to affect variables in other loops within the system.

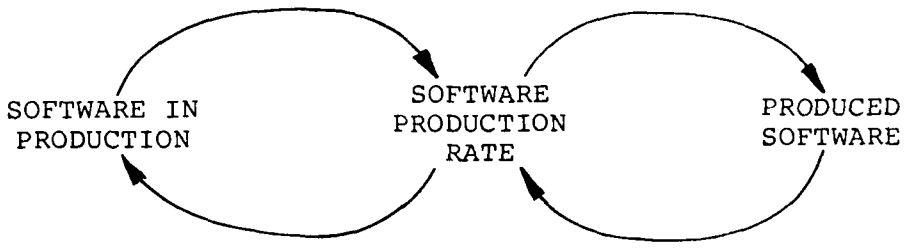


Fig. 3-3. Simple Feedback System

The last element of a causal loop diagram, as shown in Figure 3-4, is a depiction of the nature over time of the relationship between variables. This relationship is referred to as a positive or negative linkage, and is represented as a "+" or "-" at the head of the arrow linking two variables. A "+" near the arrowhead indicates that the variable at the tail of the arrow and the variable at the head of the arrow change in the same direction. If the tail variable increases, the head variable increases. If the tail decreases, the head decreases.

The "-" near the arrowhead indicates that the variable at the tail and the variable at the head of the arrow change in opposite directions. If the tail increases, the head decreases. If the tail decreases, the head increases.

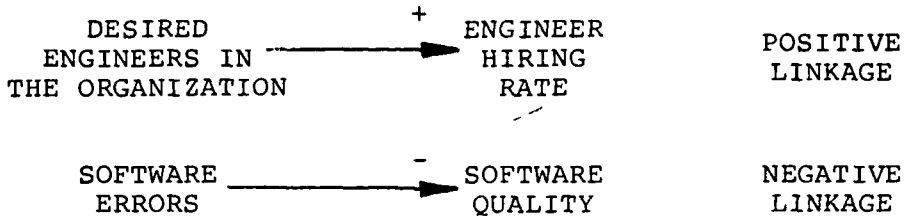


Fig. 3-4. Directional Linkages

Just as linkages have two possible directions, feedback loops have two polarities, positive and negative. The "(+)" symbol in the center of Figure 3-5a indicates a loop that acts to reinforce variable changes in the same direction as the change, contributing to sustained growth or decline. The loop in Figure 3-5a is termed a positive feedback loop. The "(-)" symbol in the center of Figure 3-5b indicates a loop that acts to resist or counter variable changes, thereby pushing to a direction opposite a change, and contributing to fluctuation or to maintainance of the equilibrium of the loop.

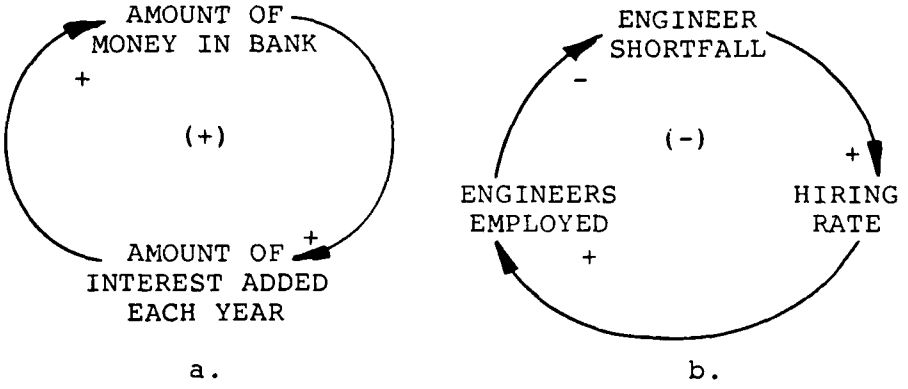


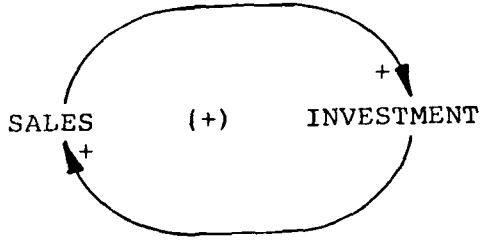
Fig. 3-5. Polarity of Feedback Loops

A simple method exists for determining the polarity of a feedback loop. The negative linkages in the loop are counted. An odd number of negative links indicates a negative feedback loop. Zero or an even number of negative links indicates a positive feedback loop.

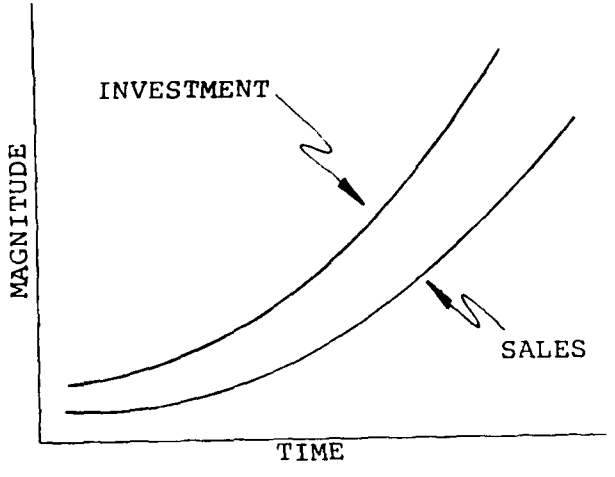
Feedback Loop Behavior. As stated previously, the most fundamental premise underlying system dynamics is that dynamic behavior is a consequence of feedback structure. Positive feedback loops exhibit a very predictable behavior in response to a change induced in any of their variables. Such loops can act only to reinforce or accelerate that initial change. Figure 3-6 illustrates this behavior. In Figure 3-6a an increase in sales generates increased investment which leads to more sales and further investment. Unless constrained by other feedback loops in a feedback system, positive feedback loops can only produce exponential change as graphed in figure 3-6b.

However, this exponential change may occur in either an upward or downward direction. Upward change indicates accelerating growth as in Figure 3-6b. Downward change indicates accelerating decline as in Figure 3-6c; when sales fall, causing investment to fall, leading to further decline in sales, etc.

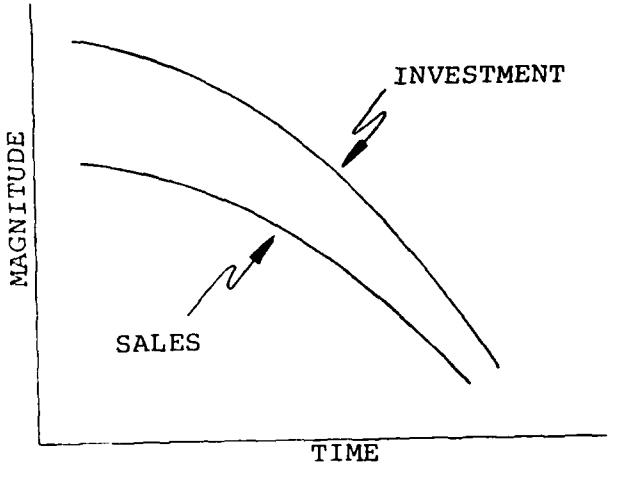
Postive feedback loops generate exponential change in all of their variables. This exponential growth or decline would continue forever, unless and until affected



a.



b.



c.

Fig. 3-6. Positive Feedback Behavior

by input from outside the loop, which, in the real world, always eventually occurs (Ref 33:16).

Negative feedback loops exhibit greater behavioral variety than positive loops. In all cases, a negative loop acts to counter the direction of initial change in any of its variables, but different forms of negative loop behavior exists.

The simplest form of negative feedback loop behavior is gradual adjustment without oscillation or fluctuation, as shown in Figure 3-7. Equilibrium is eventually reestablished after a change in a loop variable.

The second form of negative feedback behavior is called *damped oscillation*. In this case the fluctuations about an equilibrium position gradually fade away as in Figure 3-8. Such behavior is produced by more complex negative feedback loops.

The characteristic *undershoot* and *overshoot* (i.e., *a search for equilibrium*) of Figure 3-8b is found in many if not most negative feedback loops. It results from multiple delays and multiple variables reflecting accumulating information and decision processes in the negative loop. By adjusting the decision rules and/or delays in a loop it is possible to produce stable or explosive oscillatory behavior, rather than damped. Figure 3-9 illustrates these situations, which occur less frequently in the real world (Ref 33:17-18).

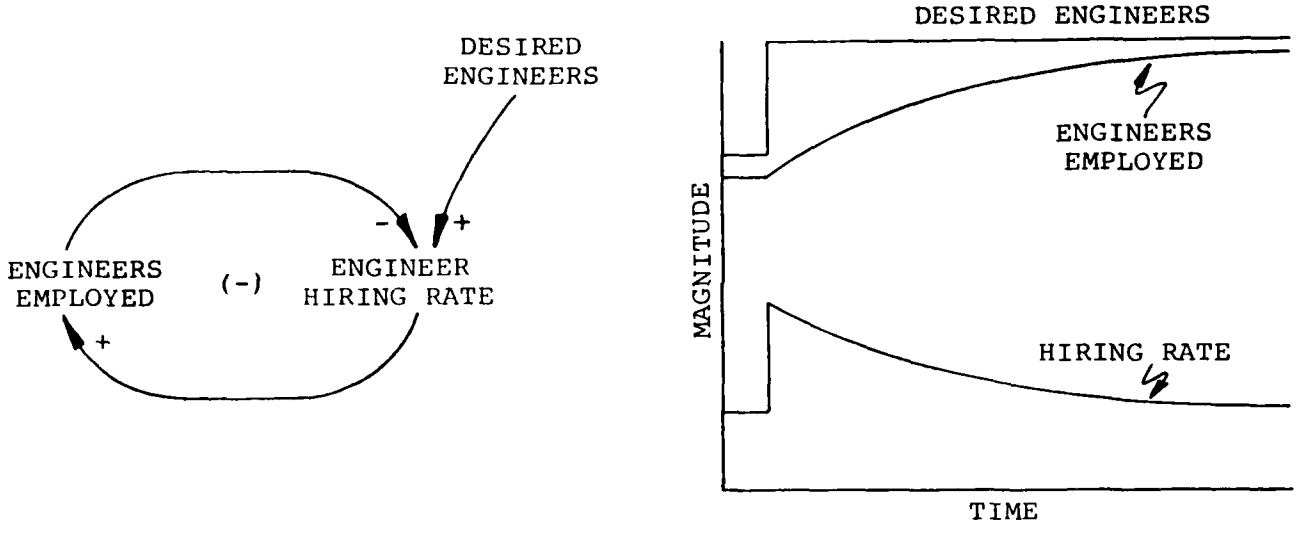


Fig. 3-7. Negative Feedback Behavior, Adjustment Without Oscillation

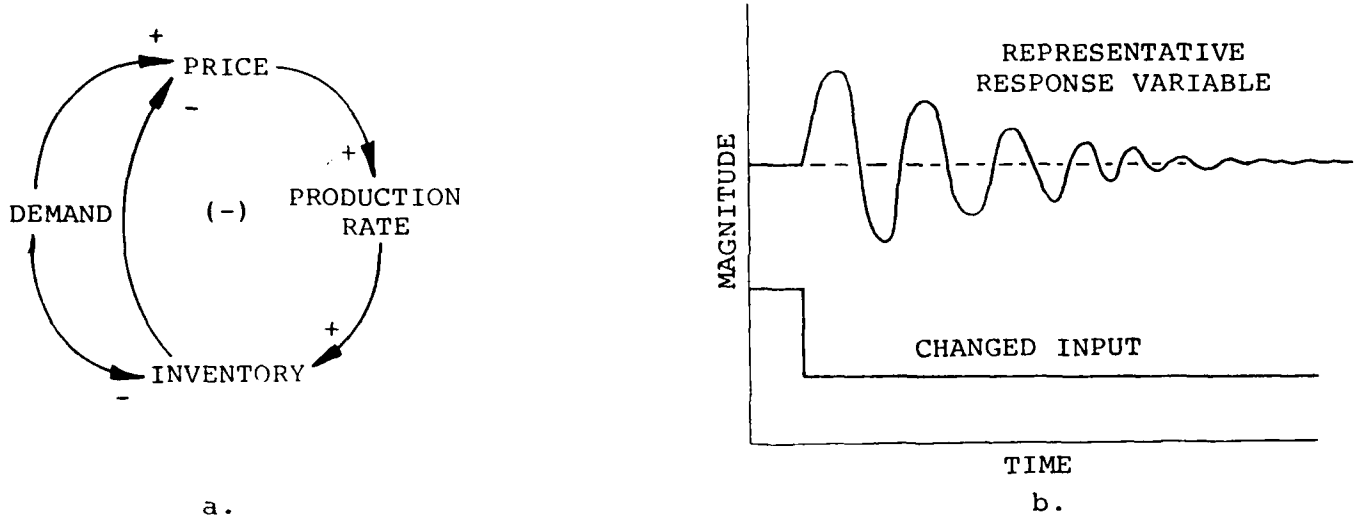
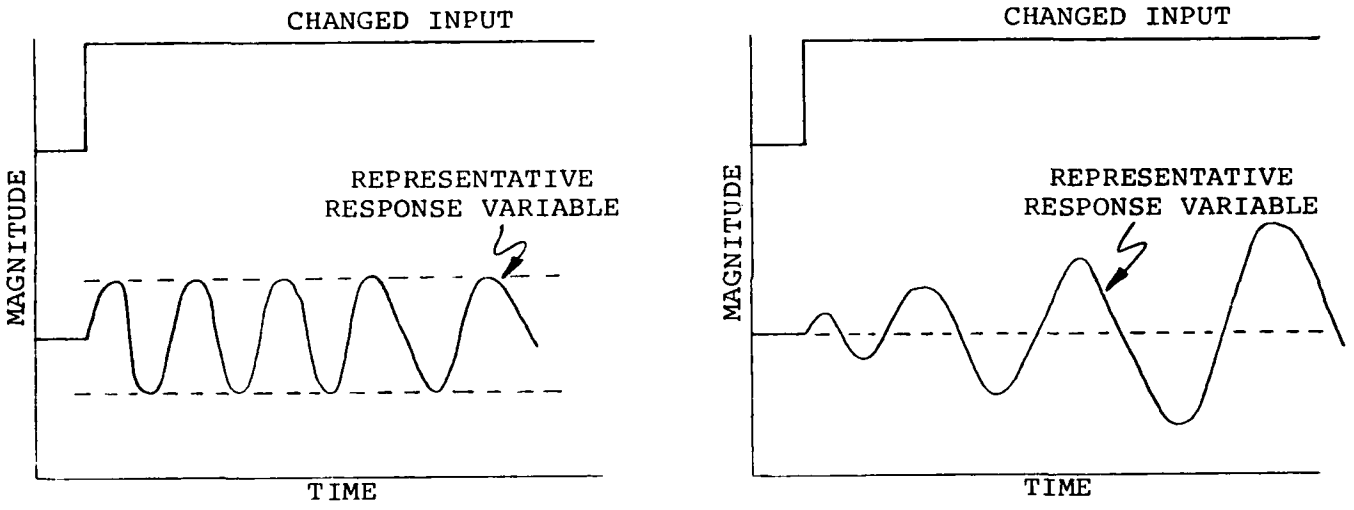


Fig. 3-8. More Complex Feedback Loop, Exhibiting Damped Oscillatory Behavior



a. Stable Response

b. Explosive Response

Fig. 3-9. Stable and Explosive Oscillatory Patterns of Negative Feedback Behavior

The behaviors described in the preceding discussion are simplistic representations of feedback loop behavior, both positive and negative. Behaviors encountered in actual managerial issues are characteristically the result of more complex, high order, nonlinear, multiloop feedback systems.

System Dynamics Flow Diagramming

Flow diagramming from a system dynamics viewpoint, provides a means for translating the feedback structure of the system being studied, as represented by causal loop diagrams at the higher levels of the structural theory hierarchy, into a more formalized and complex representation of levels and rates at the lower and more detailed levels of the hierarchy. The symbology used in flow diagramming is depicted in Figure 3-10. In Figure 3-11, the flow diagramming symbols have been combined in an example translation of a causal loop diagram.

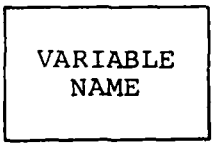
As the system model takes form the flow diagram becomes more informative. This detailed representation logically leads to the development of system equations for model computerization. A one-to-one correspondence develops between flow diagram symbols and model equations.

Equation Writing

The formulation of mathematical equations is the transition step between flow diagramming and computer

LEVELS

Measurable quantities or accumulations within the system which determine the system state



FLOWS

The movement of:

Information

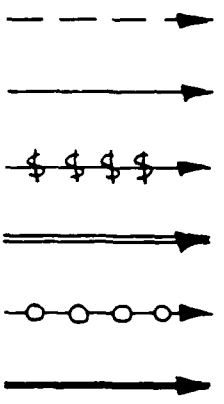
Material

Money

People

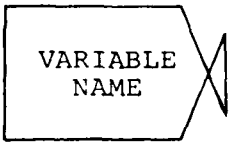
Orders

Equipment



RATES

Policies that control the flows between levels



SOURCE/SINK

Represent levels outside the system

AUXILLARY VARIABLES

Provide clarification of the policy substructure of rates

CONSTANTS

Constant system parameters

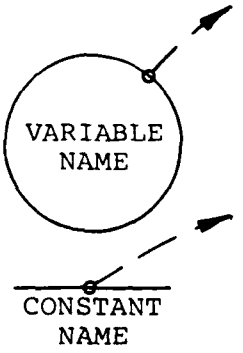
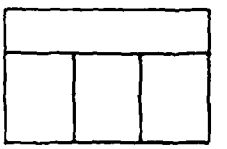


Fig. 3-10. System Dynamics Flow Diagram Symbols

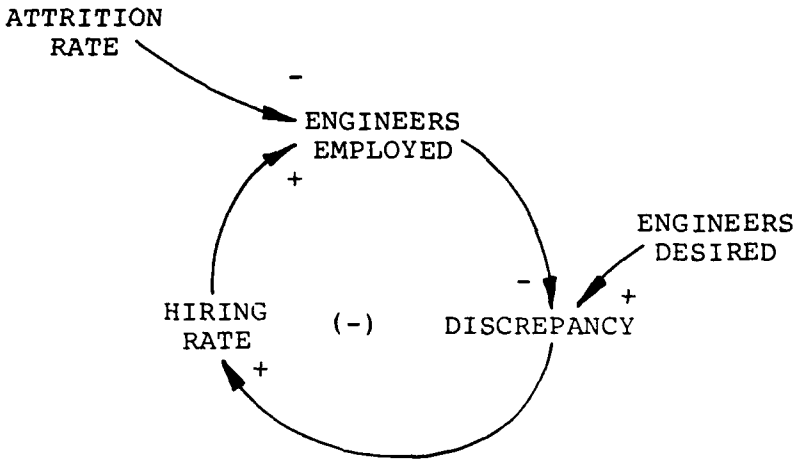
DELAYS

Describe the process of time delays

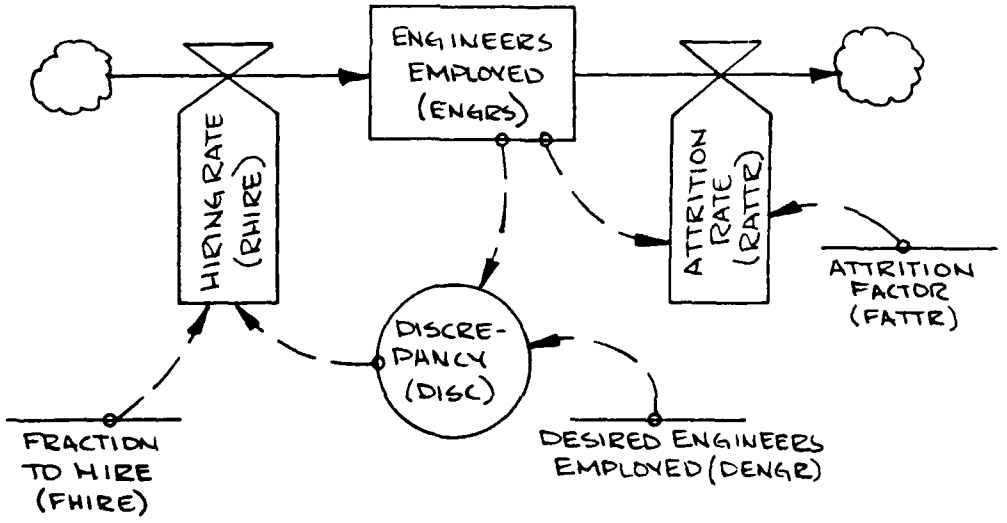


Connections to or from other diagrams

Fig. 3-10--Continued



a. Causal Loop Diagram



b. Flow Diagram

Fig. 3-11. Causal Loop Diagram and Translated Flow Diagram

simulation of system behavior. The equations become the basis for the DYNAMO language computer model.

Six equation types are used in this study: levels (L), rates (R), auxillary variables (A), initial values (N), constants (C) and tables (T). Levels, rates, and auxillary variables exhibit time dimensions. The notation used to denote time is similar to subscripting variables. The letters "J", "K" and "L" when attached to a variable name denote past, present and future time. Because rates represent movement over time, rate variable names employ a double subscript: "JK" for time from past to present and "KL" for time from present to future. As an example from Figure 3-11, "RHIRE.JK" is the model variable representing the hiring rate over the interval from past to present.

Level equations describe system state at each point in time. In Figure 3-11, "ENGRS.K" is the model variable representing the number of engineers employed at present. The value of a level at present is a function of the past level and the rates of inflow and outflow over the last time period (DT). For example:

```
L  ENGRS.K=ENGRS.J+(DT)(RHIRE.JK-RATTR.JK)
```

Rates are normally specified as functions of levels or auxillary variables, but can also be described as a constant value. In the example below, the engineer hiring rate (RHIRE) is equal to a fractional value (FHIRE)

of the discrepancy between the desired engineers employed and the actual engineers employed.

R RHIRE.KL=FHIRE*DISC.K

Auxillary variables provide clarification of the policy substructure of rates. They represent informational concepts that provide inputs to other auxillary equations or to rate equations. In the preceding equation derived from Figure 3-10, the hiring rate (RHIRE.JK) was a function of the discrepancy (DISC.K). The discrepancy is computed by an auxillary equation that compares desired to actual levels of engineers employed:

A DISC.K=DENGR-ENGRS.K

Initial value equations provide required starting values for all levels, and for rates and auxillary variables desired initialized by the modeler. If the initial value of the engineers employed was 100 then:

N ENGRS=100

Constants and tables set the policy parameters for operating the model simulation. They represent the constant and conditional policy inputs to the modelled system. For example, if the desired number of engineers throughout a simulation was 150, the following equation would be used:

C DENGR=150

If the desired number of engineers varied during the simulation as a look-up in a table, the following equation may be used:

T DENGR=80/90/100/130/150/170/200

Chapter Summation

The philosophy and techniques of system dynamics provide a prospective for viewing complex social systems. The theoretical basis for relating behavior to structure and policy provides a scientific framework for systematically arriving at system improvements. Rather than attempting to hold system behavior to a course that it does not seem to *want* to pursue, a tool is available to determine an appropriate underlying structure to provide behavior that holds itself in line with goal.

IV. The Traditional View of the Weapon System Software Acquisition and Support Process

Introduction

The traditional view within the Department of Defense, the *defense industry*, and the Air Force of the acquisition and support process for weapon systems software is embedded within a three-tiered framework of models known as the acquisition life cycle. This view has evolved within the defense bureaucracy over many years of successes and failures in weapon systems acquisitions to be perceived as adequately describing that process. The model is formalized, instituted and supported by a multitude of regulations and other publications (Ref 36).

Acquisition of Major Defense Systems

At the highest level of aggregation major defense system status is assigned to a weapon system by the Secretary of Defense or his deputy on the basis of estimated research, development, test evaluation, and production costs, and *national urgency*. Such designation would typically include such systems as the B-1, F-16, E-3A, and other aircraft weapon systems. The process model at this level is known as the Acquisition Life Cycle and is comprised of five sequential work phases: conceptual,

validation, full-scale development, production and deployment. The operation and support of embedded computer systems is implicit in the deployment phase.

As a weapon system and its associated embedded computer system develops from concept to deployment various decision points are injected into the acquisition life cycle. Each major defense system under development is reviewed by the Defense Systems Acquisition Review Council (DSARC) following each of the first three phases, after which a favorable decision by the Secretary of Defense is required for the acquisition to proceed into the next phase. These decisions are termed as the program decision, the ratification decision, and the review decision. The intent in establishing such decision points, also called DSARC milestones, was to identify the satisfactory or unsatisfactory progression and predicted progression of programs, and to permit the Secretary of Defense to redirect or terminate programs in trouble, without total loss of planned investment (Ref 35:8).

The agendas of each DSARC review differ significantly because the system under review changes during development, however all DSARC reviews have stated common objectives. These include assurance of a continuing operational need for the weapon system, adequate system performance, acceptable cost, and favorable cost-effectiveness relative to other alternatives. The anticipated agenda of

each DSARC review strongly influences the work accomplished in the life cycle phase that culminates in that review (Ref 35:8).

Conceptual Phase. The conceptual phase has two expressed primary goals. The first is "to explore, formulate and evaluate possible requirements for a new or significantly improved major defense system." Second, if the need is perceived as great enough, an "optimum, affordable, and cost effective preferred approach" to system development, production, and deployment is prepared for DSARC and Secretary of Defense review. Supporting the second goal, considerable preliminary analysis and design of software may be accomplished. This software effort is intended to be limited in level and scope to that necessary to establish technical feasibility and credible estimates of cost and development time (Ref 35:11).

The major product of the conceptual phase is an Initial System Specification stating the system's overall functional, performance, interface, design, and testing requirements for both hardware and software (Ref 35:18).

The conceptual phase has no prescribed time limit, but is normally considered to end with the Secretary of Defense's program decision to proceed into the validation phase (with or without specific program redirection), or his decision to end the program (Ref 35:19).

Validation Phase. The validation phase also has two expressed primary goals. The first is to assess or validate the conceptual phase's preferred system design approach against the system requirements as stated in the initial system specification, also a conceptual phase product. If the preferred approach proves unsatisfactory, "a reasonable effort [is] made to rectify it or to develop and validate a better" approach. A design competition open to industry is the usual approach to fulfilling this goal (Ref 35:20).

If and when a system design approach is validated, the second goal is to establish "sound, technical, contractual, economic, and organizational bases" for full-scale system development. The major product of the validation phase is a multiple design level hierarchy defining hardware and software configuration items and their functional interfaces. A system design can seldom be validated unless first developed in considerable detail (Ref 35:20-9).

The validation phase is considered to terminate with a second DSARC review and a ratification decision. A favorable decision is based upon a reassessment of the continued importance and cost-effectiveness of developing the planned system along with a validated system design approach (Ref 35:35).

Full-Scale Development Phase. The full-scale development phase is intended to yield four products:

1. A working prototype[5] of the major defense system (or *the* system, if multiple units of the system are not required);

2. Test results verifying that this prototype can meet its functional and performance requirements;

3. A cadre trained in system operation and maintenance; and

4. The documentation needed to begin the system production phase (if any), or otherwise needed for its deployment phase (Ref 35:36).

Implicit in these products are the tasks of completing the system design, resolving uncertainties, outstanding issues, and other problems, and thoroughly testing the function and performance of the prototype system and its components against the evolved system specifications. For system software, the full-scale development phase is intended to produce the initial operating versions of the software, not prototypes (Ref 35:36).

A third DSARC review and the Secretary of Defense's review decision terminate the full-scale development phase. Similar to earlier decisions the review decision may terminate the program, redirect it, or allow it to

---

[5]"Prototypes" are considered to be preproduction equipment. Their "form, function, and fit" are designed as identical to production units, but such equipment may differ from production units in other ways (Ref 35:36).

proceed as planned into the production and deployment phases (Ref 35:44).

Production Phase. The primary goal of the production phase is to produce and install in proper working order all acquired units of the major defense system. In the case of system software, the production phase is more correctly termed a replication phase. During full-scale development the operating versions of operational and support software are accepted. Replication typically consists of copying the machine-readable storage media and reproducing its documentation, both trivial operations compared with hardware production. In hardware production a more orthodox manufacturing process is employed, during which individual units are assembled and accepted over an extended period of time (Ref 35:45).

A formal transfer of program management responsibility, known as PMRT, from the acquiring command to the using command; and transfer of responsibility for support of the system to the supporting command typically terminate any production phase. If no production phase is planned, these events occur shortly after a favorable review decision. Their occurrence initiates the deployment phase (Ref 35:46).

Deployment Phase. The primary objective of the deployment phase is effective use of the major defense

system and maintenance of it in proper operating order until retired, replaced or "legitimately consumed [in] warfare." Software support is implicit in the deployment phase and consists of maintenance and modification. Software maintenance consists of investigating alleged software errors and devising corrections or *work-arounds*. Software modification involves altering software to support changed operational system requirements, or to make desired improvements. Both may involve testing of changes (Ref 35:45).

Less Elaborate Acquisitions

Acquisitions that do not satisfy the criteria for major defense systems are classified as less-than-major systems. Department of Defense and Air Force regulations allow such acquisitions to be managed less elaborately than major defense systems. However, the same ". . . management principles . . . are applicable to all programs [Ref 14]."

> Most systems consisting only of software and many smaller systems that include both hardware and software, would fail to qualify as major defense systems, at least on the grounds of predicted costs [Ref 35:47].

In terms of system software, most less-than-major system acquisitions are *managed* very similarly to major defense system acquisitions.

> Unless the software is extremely simple, little can be eliminated [from the acquisition process] without risking misunderstood requirements, incorrect operation, and loss of [support] capability [Ref 35:47].

The Computer Program Life Cycle

Air Force Regulation 800-14, Vol. II (Ref 11), defines a computer program life cycle distinct from the major defense system acquisition life cycle. The computer program life cycle consists of six work phases: analysis; design; coding and checkout; test and integration; installation; and, operation and support. These phases are perceived to occur mainly in sequence; but may loop between any and all phases, as well as overlap. A computer program life cycle "may span more than one system acquisition life cycle phase, or occur [with]in any one phase [Ref 11:2-3]." Figure 4-1 (Ref 35:7) presents the relationship between the computer program life cycle of an embedded software system, its parent system's acquisition life cycle, and a representative life cycle for the associated embedded computer system hardware.

Chapter Summation

The traditional view of the weapon system acquisition and support process, by dividing such efforts into phases of work, emphasizes the discrete nature of that work. Indeed, project life cycles can be viewed, at least after the fact, as having been composed of such segments. However, the dynamic feedback nature of the process is distorted or ignored.

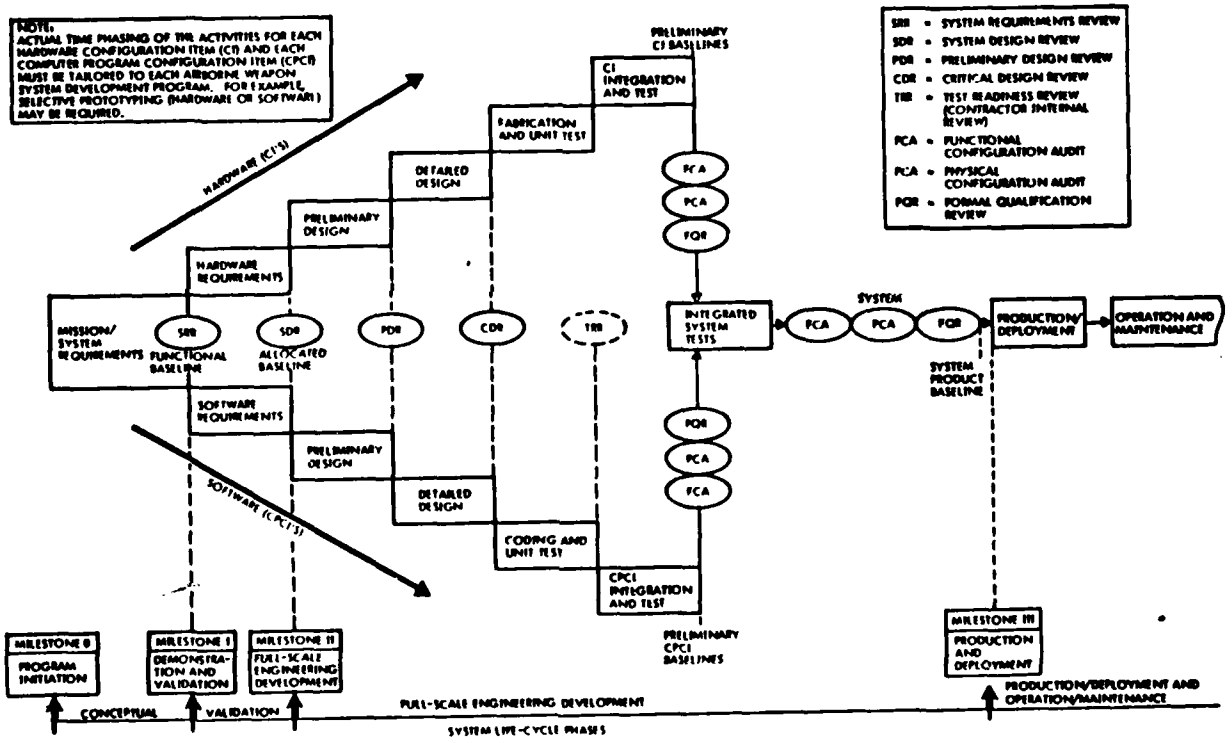


Fig. 4-1. System Life Cycles

Such a view provides useful accounting information for post-phase analysis, but no real information for maintaining a decision and control apparatus. The injection of decision points into the acquisition life cycle is an effort to utilize information available, although too often too late, for just that purpose.

V. A View of the Weapon System Software Acquisition and Support Process Based Upon a Theory of its Structure

Definition of a Context for Discussing Embedded Computer Systems

Using causal loop diagramming and an associated verbal description, the role and importance of embedded computer system (ECS) acquisition and support can be examined in a larger structural context. Figure 5-1 presents a causal loop diagram of that context. The ECS acquisition and support system is shown in the lower portion of that feedback system.

Initiation of an embedded computer system acquisition results from a funded need. In the diagram a need for an embedded computer system is translated into budget and thus a funded need.

The need for an embedded computer system results from a somewhat unknowing application of Ashby's cybernetic law of requisite variety[6] and is manifested through two sources. The first is a perception by planners that increasing complexity and difficulty in accomplishing a

---

[6]This law essentially states that "only variety can absorb variety," meaning that the variety of actions tendered by one component of a system can only be counterbalanced by an equivalent variety of responses by another (Ref 4 : 279).

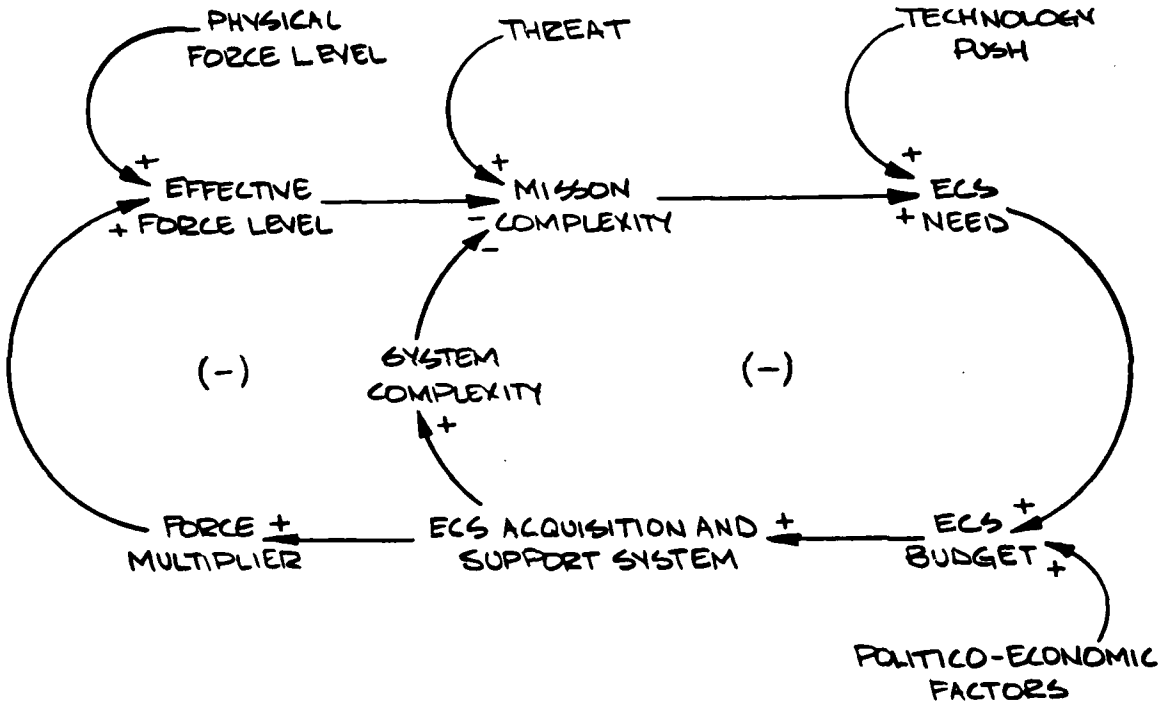


Fig. 5-1. Contextual System Environment for Embedded Computer Acquisition and Support

mission, as a result of increasing variety presented in the mission situation, merits the development of concepts (ideas) for absorbing that variety and thus the difficulties. The resulting mission complexity-driven need is a statement of requirement for the use of existing or potential technology to maintain the requisite variety of the mission situation.

The second source is a perception by planners that some available technology, as a result of general research and development, is *just what is needed* to correct a developing perception of increasing mission complexity. The development of new technology inevitably leads to a perception of *need* for that technology, a need heretofore unrecognized. This technology-driven need is many times supported by the *salesmanship* of defense contractors. The subtle difference between mission complexity-driven and technology-driven need is not important when it is realized that maintaining requisite variety in the mission situation is the central importance.

The introduction of a funded need into the acquisition and support system results in the exploration, developing, production, deployment, and support of an embedded computer system, resulting in an improvement to the capability of the operational air force. The capability or effectiveness of the air force is the result of linkages through the outer feedback loop of Figure 5-1. The

capability of the force is improved by the force multiplier effect of embedded computer systems (see Chapter I). The force multiplier coupled with a physical inventory of the force results in an apparent effective force level. This level compared with a similar measure for opposing air forces or the threat, determines the complexity of the mission presented to the force. All of these measures are accomplished on a relative scale which relates present force level to past and predicted in terms of some selected baseline. As this is a conceptual discussion, the quantification of those measures and their base is assumed not to be relevant.

The concept of mission complexity is also dependent upon the saturation of the abilities of the operators of the force in interfacing with the hardware and software of the embedded computer system. This saturation is negatively linked to the degree of responsibility for mission accomplishment assigned to the embedded computer system. (This statement implies no judgement of the relative importance of the mission inputs of man or machine.): the greater the responsibility assigned, the greater the complexity of the system constructed to accomplish it, and thus the lesser the apparent complexity of the interface between the operators and the mission, as viewed by the operators.

Returning to the threat, as its capability increases due to a similar systems process, mission complexity increases; thus creating an impetus for improved embedded computer systems and reinitiating the cycle with a need.

The last exogenous factors affecting this feedback system are aggregated in Figure 5-1 as *politico-economic factors*. These factors constitute a complex decision structure, that acts upon information from within and outside of the relatively closed system of the diagram, to cause perturbations in the translating of need to budget.

The concern in this study is the examination of those parts of the system presented in Figure 5-1 that deal with ECS acquisition and support, and its generalized input and output causal factors of funded need and force effectiveness. Figure 5-2 redefines the components of that sub ystem in terms of a process flow from need to retirement for embedded computer systems.[7] The immediate influences on the process are also presented.

It is apparent from Figure 5-2 that the dominating element in the determination of the result of the process, an effective force, is the management organization monitoring and controlling the process. Each of the acquisition

---

[7]The reader may find some of the symbols in Figure 5-2 similar to system dynamics flow diagram symbols, however this diagram is for conceptual illustration only and is not intended as a formalized system dynamics flow diagram.

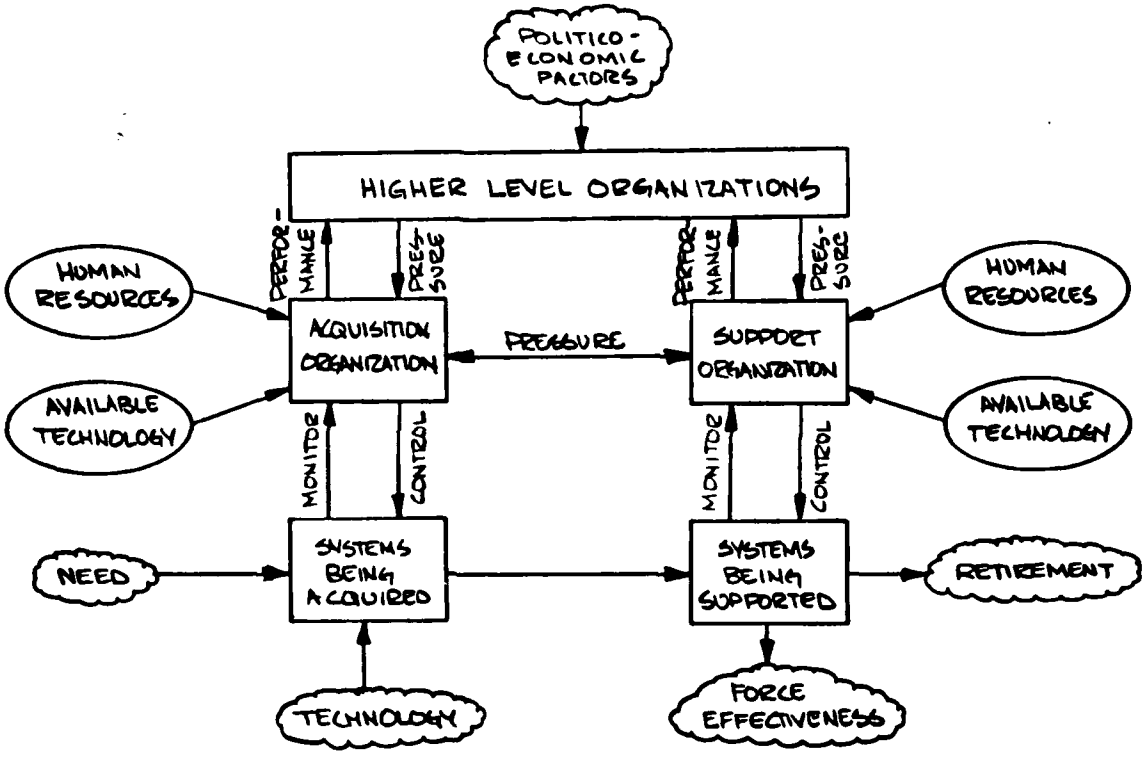


Fig. 5-2. A Conceptual View of the ECS Acquisition and Support Process and its Environment

and support components of the process is monitored and controlled by an associated organization. These organizations are, in turn, monitored and controlled by higher level organizations (DOD, OMB, and GAO, among others) influenced by among other things the politico-economic environment within which they operate. The predominant form of management control by these higher level organizations is performance pressure. Here and in later usages in this study, pressure means an increasing attempt to influence the behavior of another individual or organization. In the case of the acquisition organization this is pressure for cost, schedule and system requirement fulfillment. For the support organization this is pressure for maintenance of an effective force.

Pressure applied by higher level organizations results in an interrelationship and corresponding pressures between the acquisition and support organizations. The support organization pressures the acquisition organization for increased *supportability* of acquired systems; while the acquisition organization correspondingly pressures the support organization to accept a less *supportable* embedded computer system in the interest of *on cost, on schedule* acquisition. The establishment of life cycle costing methods is a traditional response to the perceived need to balance the pressures in the interest of reducing overall system lifetime costs. However, the immediate

costs of acquiring a weapon system and its associated embedded computer system continue to dominate the relationship.

Each organization attempts to respond to the pressures upon it and to accomplish its mission requirement (in general, *to acquire* and *to support*) by monitoring and controlling that portion of the process and resources (human and technical) within its scope of control. Essentially, the decision-making process described in Chapter III is used. Information about the acquisition and support process is monitored. This information is used for decision making, producing a controlling action for maintaining a desired result by the process.

Because the emphasis of this study is upon the software component of ECS acquisition and support, and by condensing the decision superstructure apparent in Figure 5-2, the acquisition and support process can be recharacterized as in Figure 5-3 as three parallel flows: the continuous allocation and injection of resources, the flow of the software itself from concept to development (acquisition) to operational deployment (support) to retirement, and the performance monitoring and controlling of the process.

A very important concept becomes apparent in Figure 5-3. That is: the activities of the acquisition and support process, the monitoring of those activities,

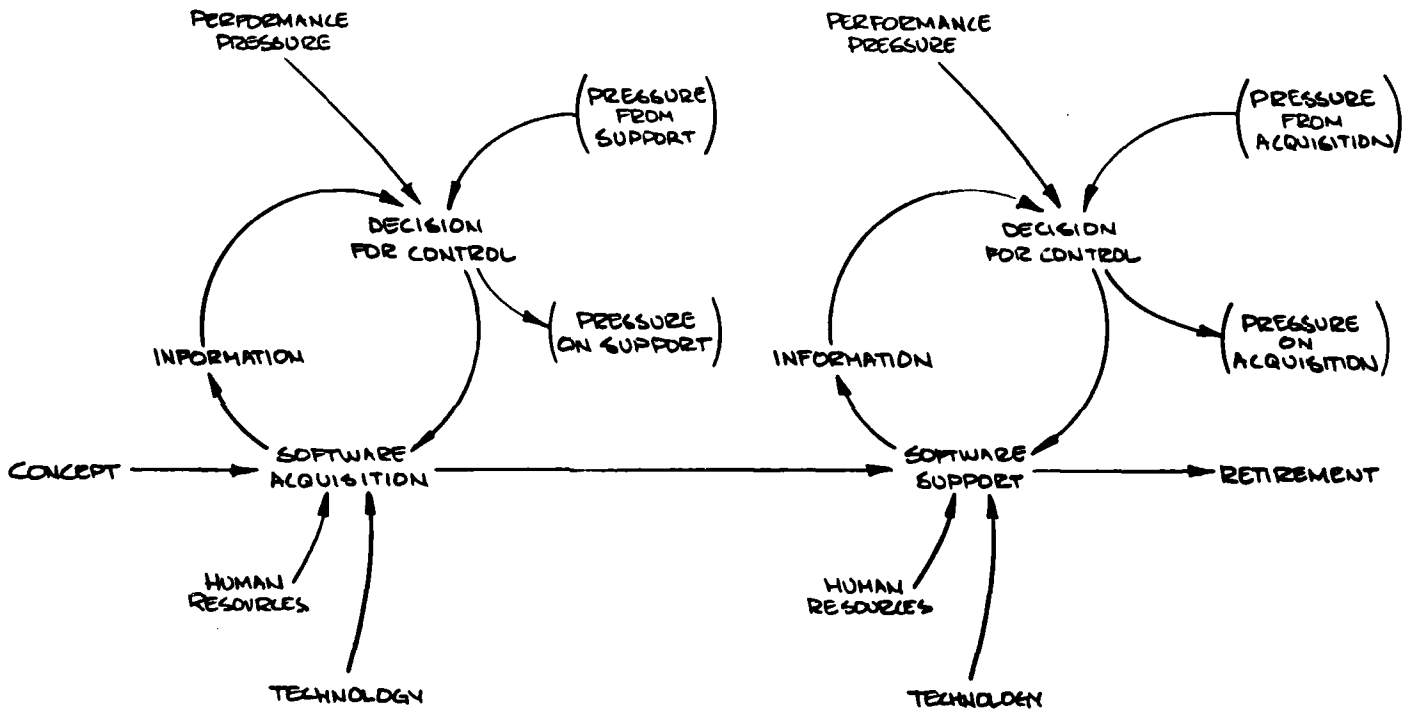


Fig. 5-3. Causal Loop Diagram of Weapon System Software Acquisition and Support

and their control form a feedback system structure. That structure is the core of a system dynamics study of the process.

Measurement of System Effectiveness

In the previous discussion the generalized output of the embedded computer and weapon system software acquisition and support systems was shown to be force, or weapon system, effectiveness. How can this variable be quantified, so that the performance of the process can be measured?

The determination of weapon system effectiveness, in conjunction with determining life cycle cost, has been an area of study during the past two decades attempting to obtain *cost-effectiveness* measures for evaluating weapon system alternatives by the defense department. A framework for accomplishing such efforts was described in the 1965 publication of the Weapon System Effectiveness Industry Advisory Committee Report. That report defined systems effectiveness as the product of availability, dependability and capability.

Availability was defined as a measure of a system's (or subsystem's) condition at the start of a mission, when called upon at an unknown point in time--available or not available--and can be expressed as a function

of uptime (time available) and downtime (time not available) where:

$$\text{Availability} = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}}$$

Uptime is a system reliability oriented measure, while downtime is a system maintainability oriented measure. All weapon systems are down for maintenance at some time, scheduled or unscheduled. More reliable systems exhibit less unscheduled maintenance than less reliable systems. The maintainability of the system determines the length of downtime.

In terms of weapon system software, reliability is a product of software production, where production is defined in a broad sense as the embodiment in computer-usable form of a concept of use and direction for such a computer. Less reliable, poorly produced software exhibits more latent errors, either errors in coding, testing or fulfillment of design requirements. Maintainability is the result of actions taken by the software designer during the design portions of production so that the system when produced, installed, and operated can be effectively maintained. Both reliability and maintainability are affected by the success of the software testing program in predicting the quality of the reliability and maintainability efforts.

Returning to system effectiveness, its second component, dependability, is a measure of system condition during a mission. A dependable system operates as specified throughout the range of its operating environment. This environment may become bounded by some overall system degradation during the mission. Excluding environmental factors, dependability is solely dependent upon system reliability.

The third component, capability, is a measure of a system's ability to accomplish a mission given its condition during that mission. This measure is a function of the modes and policies of system employment, as well as system reliability.

The overriding issues in weapon system software effectiveness are the quality of the reliability and maintainability efforts in production (including design) and testing. It was stated earlier that reliability was a function of latent errors in the software, or failure of the software to perform as required. The quality of the acquisition effort, or the level of latent errors in the produced software affects the quality of the support effort by providing an initial loading of corrections or changes to be made to the software. The quality of the support process is dependent upon the management of changes to deployed software. This change management is also known

as configuration management, or the management of the system design configuration as a result of change.

Software changes during support stem from causes in addition to latent errors. Three causes of software changes are:

1. Latent Software Errors. As defined before, such errors are the result of errors in coding, testing and requirement fulfillment during production. Those latent errors, resulting from the acquisition process, are undiscovered upon transfer of the system to the support process.

2. User Suggested Enhancements. During the operational life of the weapon system, unforeseen enhancements to improve system effectiveness become known. Such changes are the result of experience in use, or poor conceptualization, requirements definition and/or design during acquisition.

3. New Capabilities. Some software changes are mandated during the operational life of the weapon system by genuinely new roles and missions established for the system and intentional tradeoffs in requirement fulfillment to the support process during acquisition. Such tradeoffs occur due to reductions in budget or schedule either within or outside the acquisition organization.

It becomes apparent, from an analysis of the sources of software changes, that the quality of the acquisition effort predetermines some limitation to the

quality of the support effort, which in turn establishes the system's effectiveness during its operational life.

The development of quality software is the area of practice for the disciplines of software engineering and management. Figure 5-4 presents a causal loop diagram of the weapon system software acquisition and support process in the context of the software engineering and management techniques employed. The previous analysis showed that the quality of acquired software, at the point such software is transferred to the support process, initially determines the level of software changes required. The central linkage, in the diagram, between the acquisition and support processes represents this transfer.

In interpretation of Figure 5-4, software quality is monitored by the acquisition organization using a quality measurement method, generally software testing. A perception of the quality of software being acquired results. This perception is translated into some effort or pressure to maintain or attain a desired level of quality through a quality assurance method. Increasing and decreasing pressure for quality in software production results. The feedback loop continues to operate as software is developed and transferred to operations/support.

During the support process, software changes to be made in a particular weapon system software packare are determined by the three types of changes discussed

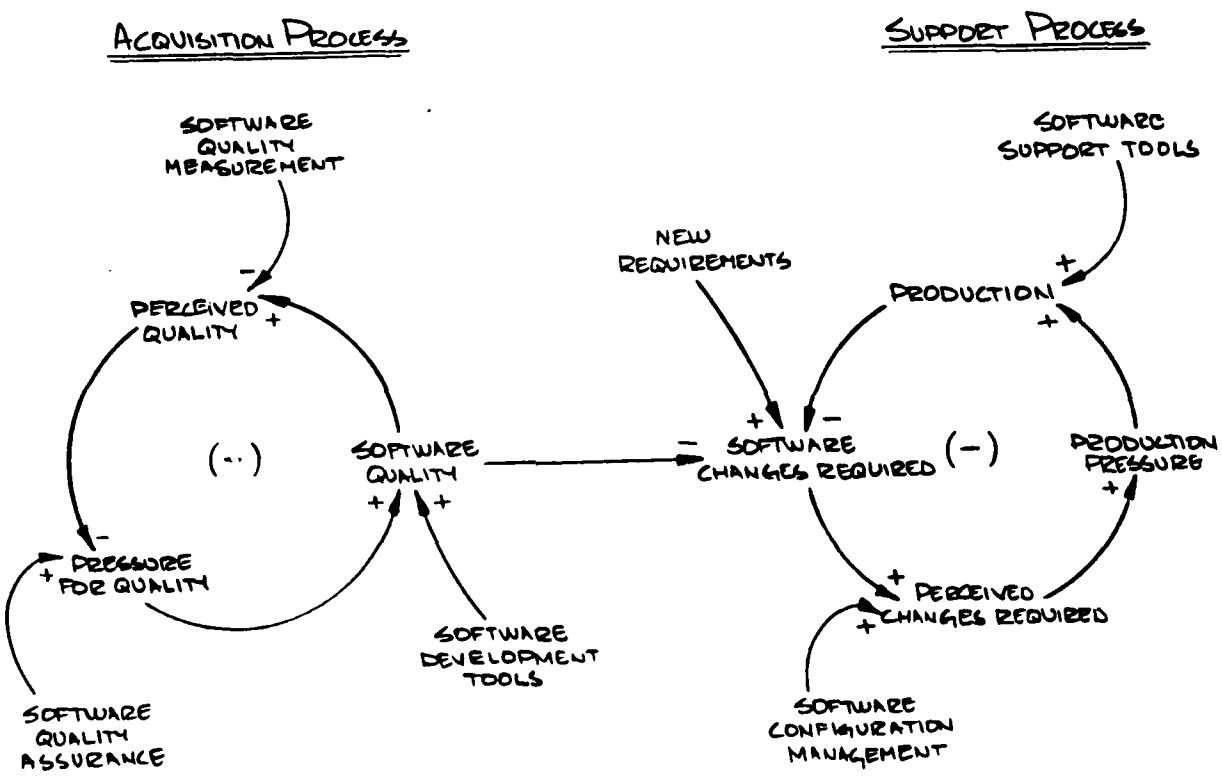


Fig. 5-4. Weapon System Software Acquisition and Support within the Context of Software Engineering and Management

previously. In the diagram, user suggested enhancements and new capabilities are aggregated as *new requirements*. The configuration or change management capability of the support organization measures and provides a perception of the changes required. This perception is translated into increasing and decreasing pressure for change production. Change production employs software support tools to decrease the backlog of changes; while new requirements continue to create new changes to be made.

Development of a Conceptual Structure

In Chapters II and IV, one stated criticism of the traditional approach to research into the framework of software acquisition and support, was the emphasis of such research upon partitioning the process into discrete sets of activities separated in time and lacking any base of common elements to bind them. We can now ask if the foregoing analysis leads to the recognition of a common base of underlying structure in the weapon systems software acquisition and support processes?

Referring to Figure 5-4, similarities do indeed exist between the acquisition and support processes. First, both operate as production processes, transforming an input into an output. In the case of acquisition, quality software is the intended product. The quality of the software is exhibited in terms of conceptualization of

an appropriate software system requirement, fulfillment of that requirement through design and coding, and the detection of errors or rework by some testing method. In the case of support, changes are made or, in different terms, quality software is maintained. In both cases, software quality can be measured by the amount of work and rework required to attain that level of quality.

Second, both acquisition and support use resources, technology (software development and support tools) and manpower, to perform the transformation or production. Both are characterized by some measurement of their production and readjustment or control of the process based upon that measurement.

From these similarities a generalized process can be described which is descriptive of both acquisition and support. In terms of a production process, some level of original backlog of work exists: software to be developed or changes to be made. This backlog is transformed into work accomplished, at a rate of accomplishment determined by the manpower available and the productivity of that manpower in terms of the effectiveness of their tools or technology. This generalized process is shown in Figure 5-5.

However, it is known that discrepancies or errors exist in the work accomplished. These errors can be described as work that must be reaccomplished or rework. The amount of this rework is determined by some rate of

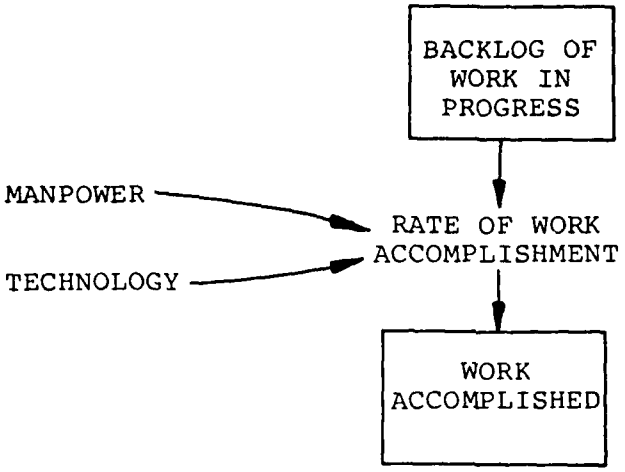


Fig. 5-5. Simplified Software Production Process

error in production. Initially, the amount of this rework is unknown or undiscovered by the process management. Only through some effort at discovering this rework, software testing in this case, does it become known and added to the backlog of work in progress. The amount of this discovered rework is determined by some rate of discovery, which itself is determined by the manpower available and assigned to the discovery task, and the productivity or technical effectiveness of their tools and techniques for discovery. Figure 5-6 illustrates the completed feedback structure of this generalized production process.[8] Software quality as reflected in this model represents the fraction of work that will not require rework.

---

[8]Cooper, et al., at Pugh-Roberts Associates first proposed a similar production loop to describe naval ship production at Litton Industries in 1978 (Ref 7).

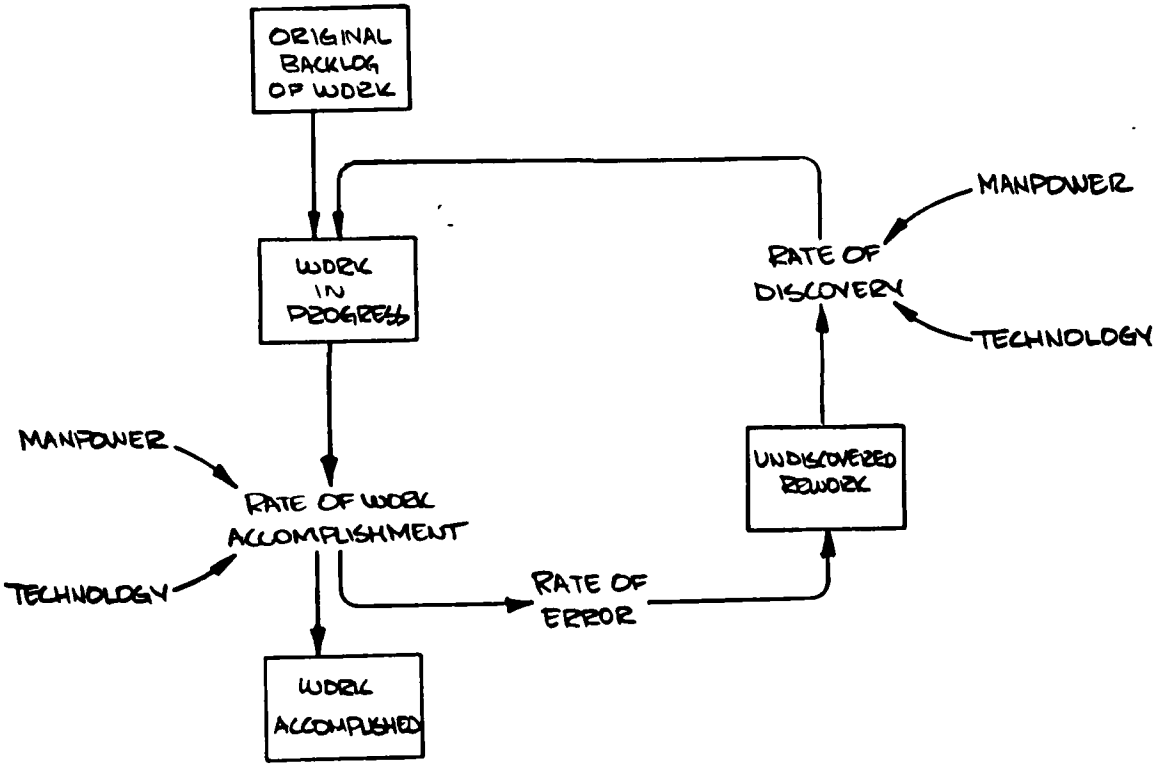


Fig. 5-6. The Generalized Software Production Loop

The generalized software production loop of Figure 5-6 represents a common underlying structure that can be used to model each of the acquisition and support processes. The level of undiscovered rework in the acquisition loop at the transfer of program responsibility from acquisition to support inversely represents the quality of acquired software, and directly represents the initial backlog of work required to accomplish software changes during support due to latent errors. The transfer of this undiscovered rework to the support loop represents an initial level of undiscovered rework in that loop. Such undiscovered rework in the support process is discovered through exercise of the operational software or by the support organization in processing new requirements for that software.

System effectiveness can be evaluated as a function of undiscovered rework and work in progress in the production loop. Undiscovered rework represents the quality in production factor in the previous discussion of measurements of system effectiveness, as such, it affects availability, dependability, and capability.

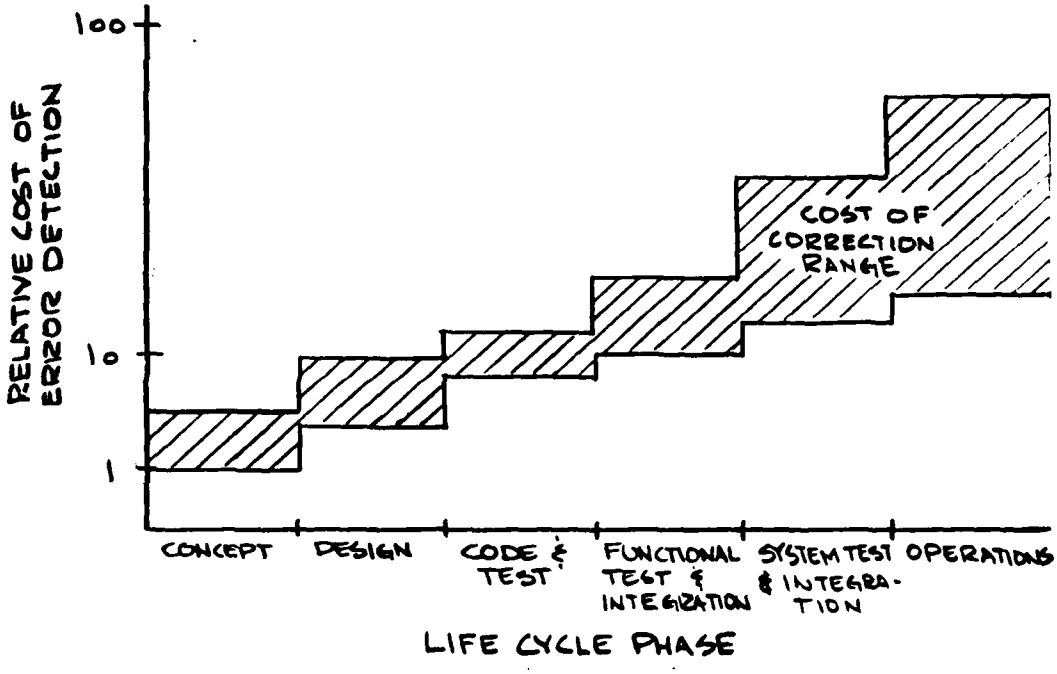
Work in progress in the support loop affects only capability. Until a new or desired capability, as reflected in a new requirement, is added to a weapon system software package, it is a part of the backlog of work in progress. Employment of that software package in its

new role before the change is accomplished, will result in less than optimal capability.

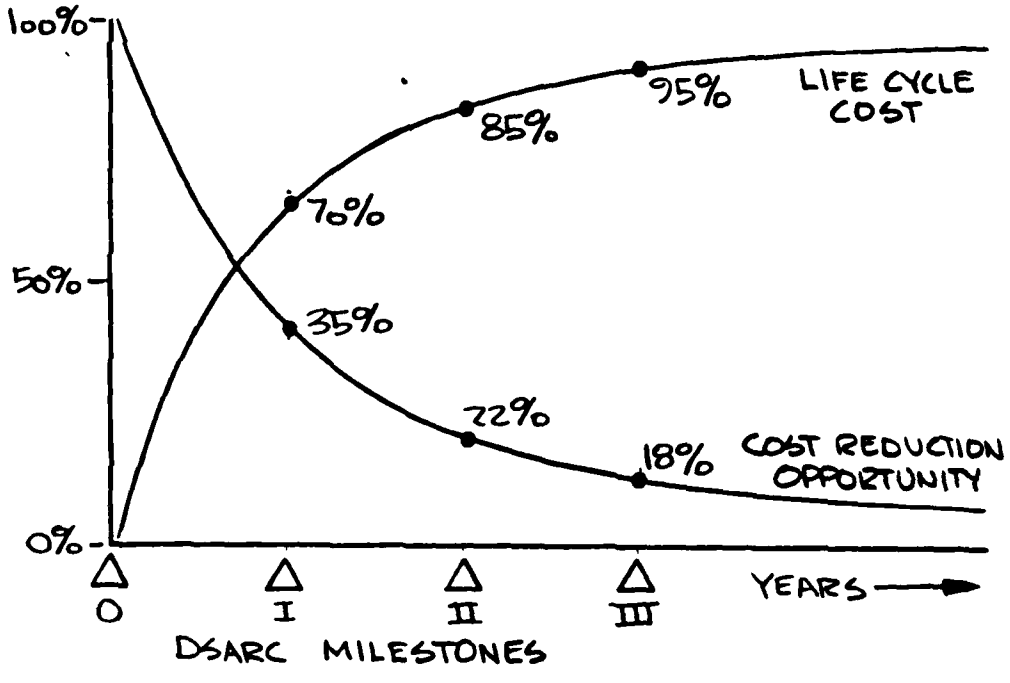
The same production loop can be used to describe the phases or steps in the acquisition life cycle or a system level computer program life cycle, and bring to that description a dynamic component. A separate loop is apparent for each phase of either life cycle.

The support process, as discussed in Chapter IV, is the last phase (deployment) of the acquisition life cycle. Because of this a single production loop models the support of each weapon system software package.

An important point presented by a system model constructed of more loops is that the development of software quality in each phase is constrained by the quality of the effort in its preceding phases. The discovery and correction of errors becomes more costly and difficult as the acquisition continues. This phenomena is represented in the graphs of Figure 5-7. As an example, in Figure 5-7a the cost of correcting an error detected during code and test ranges from 5 to 10 times that of correcting the error during concept formulation, if possible. In Figure 5-7b, the opportunity to reduce cost at the third DSARC milestone is only 18 percent of overall cost reduction opportunity. These examples serve to illustrate that cost and production problems are amplified due to the delay in identification and correction of errors in the



a. Cost of Errors (Ref 24:58)



b. Influence of time on Ability to Curb Costs (Ref 1:25)

Fig. 5-7. Cost and Difficulty in Correction of Errors

acquisition process. This is an important result of the dynamic nature of the process.

A System Dynamics Flow Diagram of the Generalized Software Production Loop

The generalized structure of the software production loop can be easily formalized as a system dynamics flow diagram. Such a flow diagram is shown in Figure 5-8. All of the structural relationships described in the preceding theory are represented explicitly in the flow diagram. As discussed earlier, each process (acquisition and support) can be represented as a series of dynamically interacting phases. While the activities in these phases differ from one to another in many specific ways, they share a common basic structure.

In the context of system dynamics, three levels and three rates are sufficient to represent the production loop. The rate of software production decreases the level of work in progress and adds to the level of work accomplished. As work is accomplished (or software is produced) it flows to work actually accomplished or to undiscovered rework, depending upon the error rate in production. The rate of discovery of rework decreases the level of undiscovered rework and increases the level of work in progress with the addition of discovered rework.

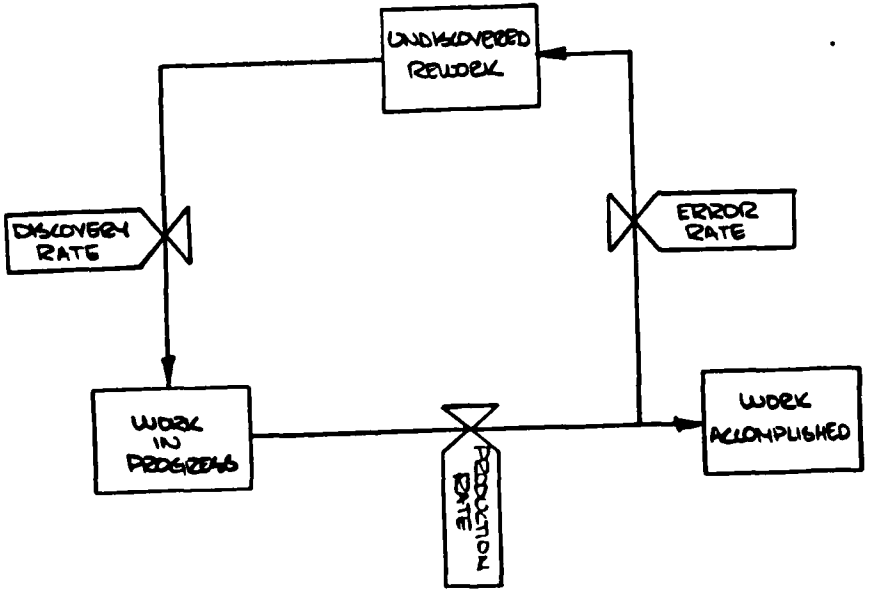


Fig. 5-8. System Dynamics Flow Diagram of the Generalized Software Production Loop

As will be seen in succeeding chapters, this simplistic system dynamics model is influenced by four general sectors: technology, manpower and productivity, production management and quality management. The production loop itself forms a fifth sector in a system dynamics model. This extended model is presented in Figure 5-9.

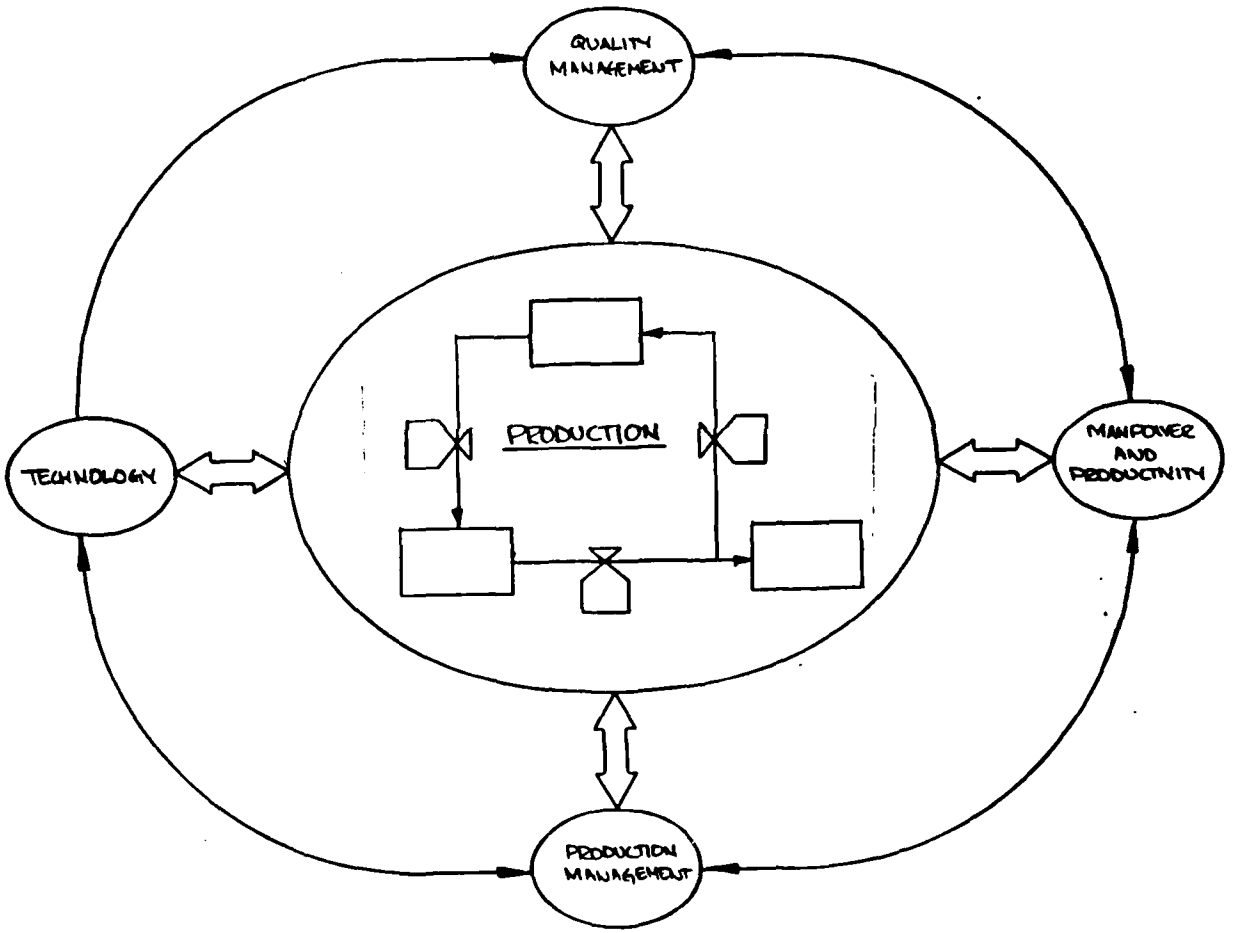


Fig. 5-9. An Expanded System Dynamics Model of the Generalized Software Production Process

VI. Conceptual Influences Upon the Structure and Behavior of the Weapon System Software Production Process

Introduction

Although the generalized production loop model is a simple representation of the software production process, the influences upon that process are many and complex. These influences operate directly upon the rate or policy components of the production loop. While the levels of the production loop are simply accumulations within the system, flows into and out of the levels are controlled by the rates. Thus, the levels are indirectly affected by these influences.

The conceptual basis and structure of these influences are presented in this chapter. As is characteristic of system dynamics models, they are grouped and described as model sectors.

Overall Monitor and Control of the Process--The Production Management Sector

In the previous chapter it was stated that the dominant influence on the software production process was the management organization monitoring and controlling the process. While the production loop is a representation of the actual process of software production and the

behavior generated by that structure is intended to represent the actual behavior of software production; the actions of management are based upon measurements or perceptions of actual behavior and structure. Such perceptions are perturbed by errors and biases in measurement.

Roberts addressed this difference between the actual and the perceived or apparent in his model of the control system structure of organizations, shown in Figure 6-1. This model is an implementation of the lowest level of the system structural hierarchy described in Chapter III.

Four characteristics of the model are noteworthy. First, the transformation of decisions into results or achievements takes place within a complex structure which is not always observable because of numerous sources of noise or random behavior and lengthy delay between cause and effect (Ref 31:393).

The second characteristic is that distinction between actual or real achievements and those apparent to the organization. The real situation is translated into the apparent through information channels which contain delays, noise, and bias (Ref 31:393).

The third aspect of the model is that decision making is represented as a response to the gap between objectives and apparent progress toward those objectives. The last characteristic is that now well-discussed

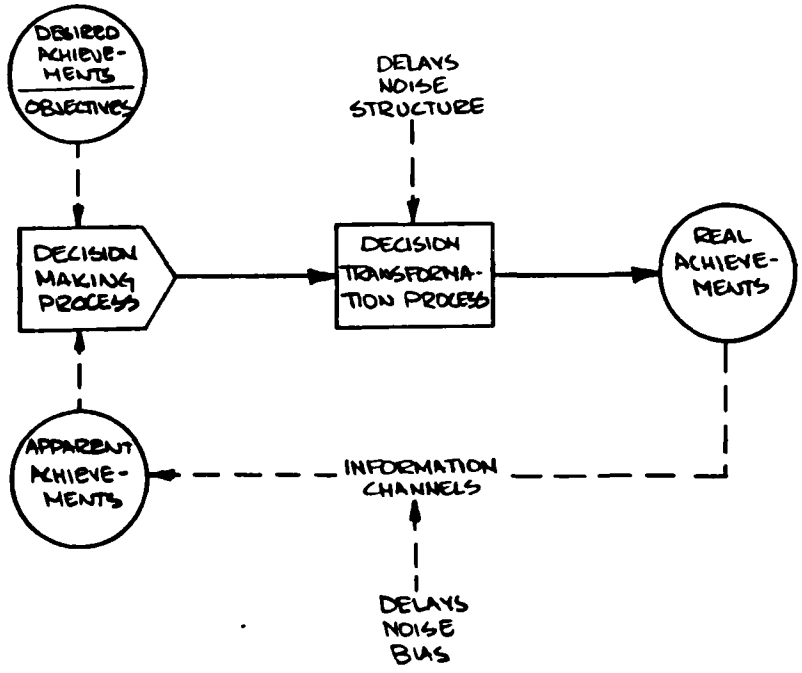


Fig. 6-1. Control System Structure of Organizations (Ref 31:394)

continuous feedback path of decision-results-measurement (information)-evaluation-new decision (Ref 31:394).

The management organization monitoring and controlling the software production process operates as a control system, just as in the Roberts' model, continually adjusting its decision making and monitoring the results in a feedback structure. The adjustments to the process are in the form of more or less pressure for production, and adjustments to schedule and performance requirements. The well-known triad of acquisition management--cost, schedule, and performance--is continually manipulated to converge towards the goal of project completion *on cost, on schedule, and within system requirements.*

Those influences on the software production process primarily concerned with measuring progress toward goal accomplishment and with adjusting or controlling overall process behavior to reach that goal constitute the production management sector. Figure 6-2 presents a causal loop diagram of the production management sector and a description follows.

Because some error, in the form of inadvertent features of the organization's information system or as chosen characteristics of that system which make such tradeoffs as accuracy for compactness, exists in the information channel translating real accomplishment to apparent or perceived accomplishment, the measurement of

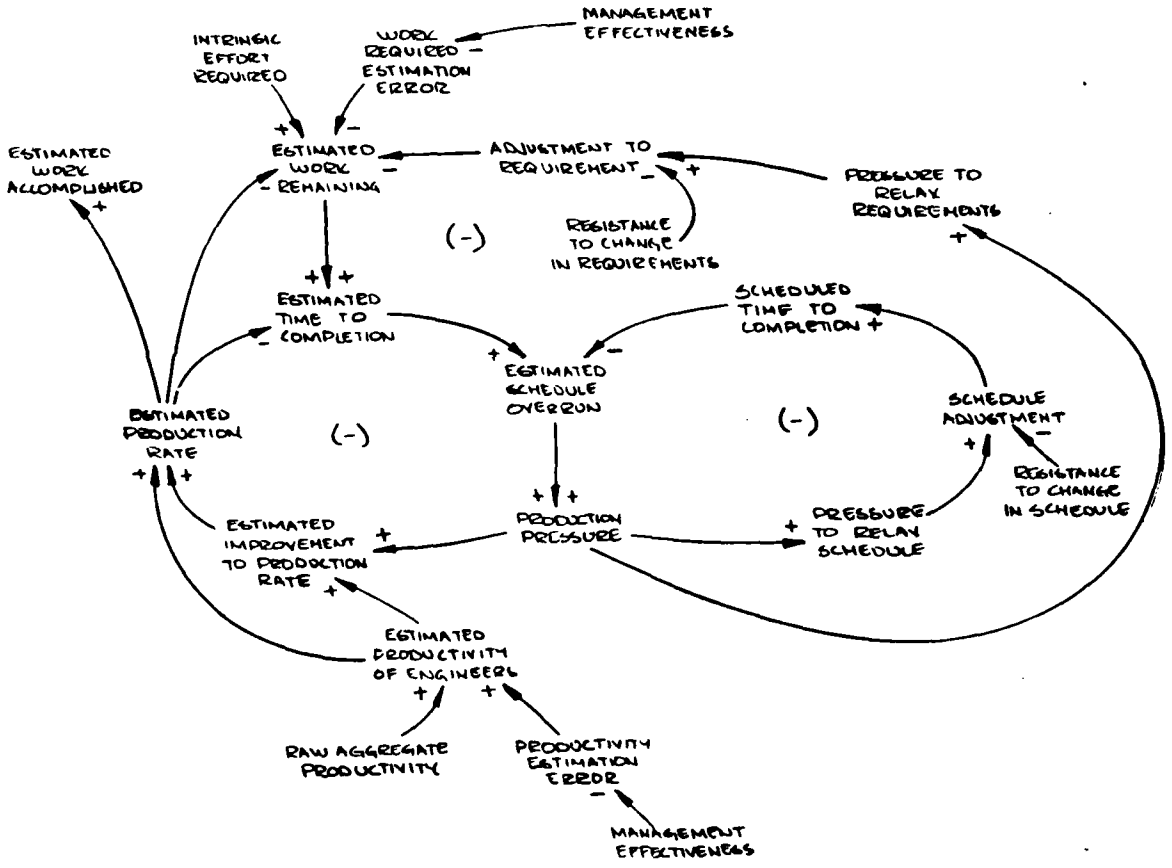


Fig. 6-2. Production Management Sector

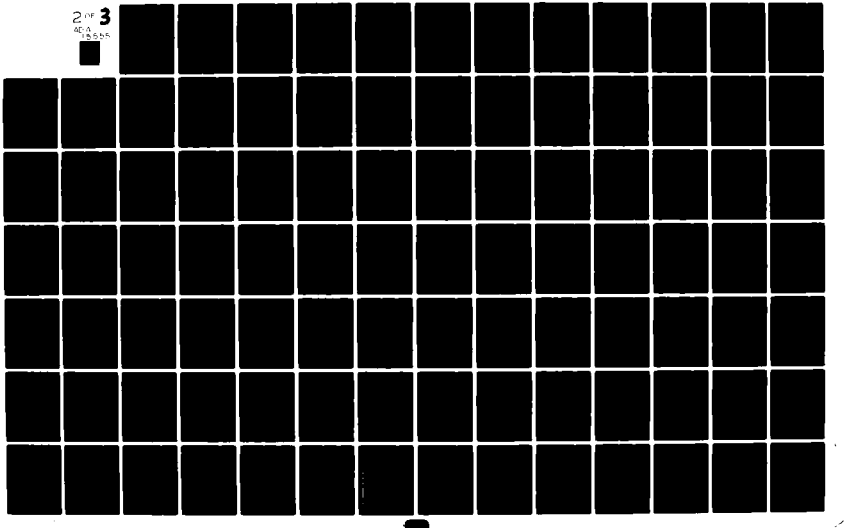
those accomplishments in an estimation. Such estimations provide the information basis of the feedback loops in Figure 6-2. These estimates can be divided into two types: initial estimates of time and effort required to complete a software project; and revisions of both estimates during the life of the project. Initial estimates are the result of viewing what may be called the intrinsic effort required to complete the project, through the distortion of an information channel.

Inherent in the production of any product, such as weapon systems software, is a set of project tasks, the magnitude of which is determined by the nature of the product. Many aspects of the product contribute to this project magnitude, including the complexity of the product. The steady trend in embedded computer systems and weapon system software has been towards increased complexity (as shown in Chapter I) thereby increasing the intrinsic project size. When compared to the technological capability available (the technological state of the art), the intrinsic size of the project is translated into the intrinsic effort required to complete the project. Of course, such abstractions are non-apparent to production management and exist only as estimations of time and effort.

The initial estimate of work remaining coupled with an estimate of the software production rate results

AD-A115 555 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYS--ETC(U)
MAR 82 B D MERCER

UNCLASSIFIED AFIT/GCS/MA/82M-3 NL



in an estimated time to completion of the project. If acceptable to the management organization, this estimated time to completion establishes the initial schedule.

Once initiated, as the production process progresses the estimated time to completion is continually compared to the schedule time to completion. From this, the degree of schedule overrun (or attainment) is determined. Process management's reaction to schedule overrun results in increased pressure for software production. This pressure is transmitted to the production process as an increase in the software production rate through the technology, and manpower and productivity sectors.

Production pressure is translated into pressure for lengthening the project schedule and for reducing project requirements (thus reducing the effort required). The combined effects of these actions are estimated by the process management to arrive at the goal of project completion on cost, on schedule, and within requirements. Within the higher levels of the organization exist levels of resistance to change in schedule and requirements. These resistances tend to keep the process operating towards accomplishment of the original project schedule and requirements.

During the life of the software project, production management provides overall monitor and control of the process through adjustment of schedule and performance

requirements, and the application of pressure for production throughout the system.

Primary Influences on the Software Production Rate--The Technology Sector and the Manpower and Productivity Sector

The Technology Sector. As presented in Chapter V, the rate of production of weapon system software is dependent upon the manpower available and its productivity, in terms of the technical effectiveness of the tools and techniques employed. The following discussion will consider the role of technology in accomplishing the production of weapon system software.

The production of any product is dependent upon the use of some set of tools and techniques. Weapon system software is no different. The engineering effort producing weapon system software is dependent upon programming languages and techniques, compilers, analyzers, translators, emulators, simulators, design techniques and hardware aids, and hardware to run, test and replicate that software. This set of tools and techniques constitutes the technology utilized.

As discussed in the previous section, the level or capability of the technology available and employed determines the time and effort required to accomplish a particular software project. Many times the undertaking of a new project brings to light the need for development

of new technologies, and the rate of accomplishment of the project is affected by the delay in advancing the technological state of the art to the new required level.

Continued research and development is accomplished in order to provide new technologies in support of new products, weapon system software included. This continuing effort hopefully reduces the delay in developing new technologies needed for project completion. The results of such efforts build upon previous efforts in continuing to advance the state of the art. The level of technology accumulates as time passes. This concept of technological growth can be illustrated for one technology as in Figure 6-3, showing the increase in number of switching gates implementable on a single integrated electronic circuit of an arbitrary size.

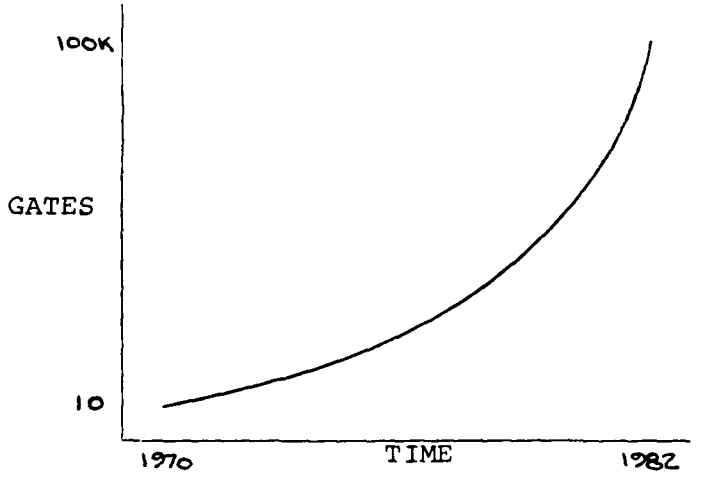


Fig. 6-3. Growth in Integrated Circuit Technology

Figure 6-3 is a typical presentation of the measurement of the advance of technology. Some parameter of a technology, in this case switching gates implementable by digital electronic technology, is graphed as a function of time. Such models typically present an exponential growth rate.

A missing ingredient in such a model of technology is the effectiveness in utilizing or implementing a particular technology. In highly complex large-scale technological projects, such as weapon system software production, that effectiveness is greatly dependent upon the management of the project or the capability of organizational management to support the implementation of a required technology.

Embedded computer projects typically operate at the forefront of technology. The management tools of the last project may not be sufficient to support the next. In fact, many embedded computer program managers maintain that they don't really know how to manage a project until it is completed and observed in retrospect (Ref 21). This leads to the addition of a second curve to a generalization of Figure 6-3.

In Figure 6-4 the improvement of a technological parameter is shown as before; however, a second curve shows the improvement in management capability to implement differing levels of parameter capability. Two important

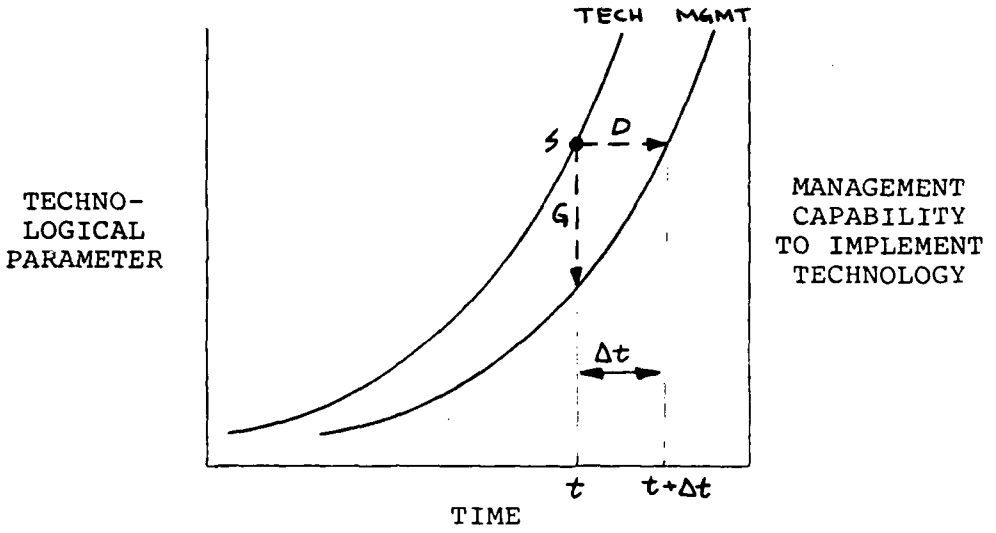


Fig. 6-4. Management Capability to Implement Technology

concepts are apparent from this figure. First, at any time, t, there is some gap G in management capability to implement the technological state of the art S. Second, there is some delay $D=\Delta t$, in the improvement of management capability to fully implement a given level of technology.

The selection of curves in Figure 6-4 presents a somewhat misleading view that management capability may never catch up with the state of the art in technology and that the gap may be ever widening. In reality however, this does not seem to be the case. Integrated technologies, as opposed to single technological parameters, appear to develop in generations. Generations in computer technology are a good example. Figure 6-5 shows how a generation of technology grows and then levels off as the

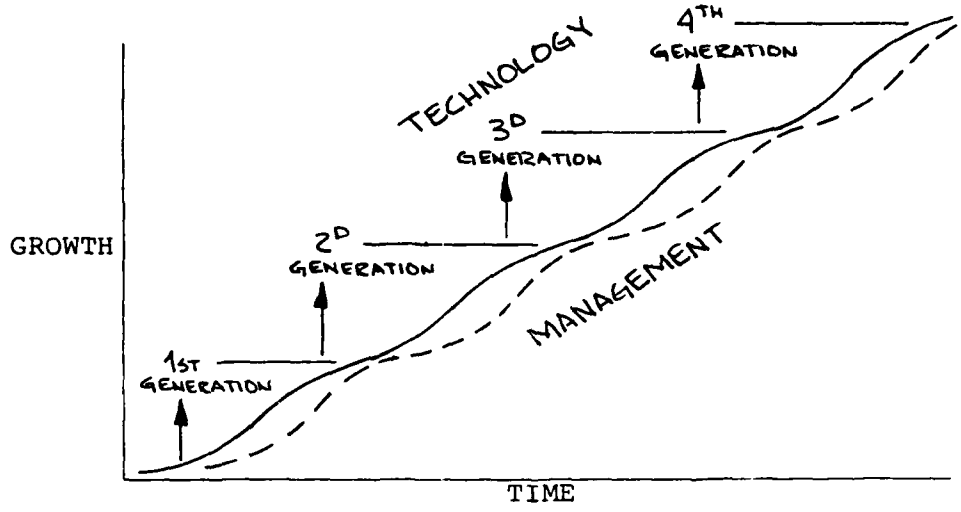


Fig. 6-5. Technological and Management Growth in Generations

limitations of its tools, techniques and devices are approached. However, management capability to implement the technology continues to grow until it catches up. Perhaps unfortunately for technology managers, new generations of technology continue to develop, renewing the cycle.

Although the technological state of the art may not be an adequate measure of the technological effectiveness of an organization involved in high technology projects, such as the production of embedded computer systems; by coupling that state of the art with the gap in management capability to implement a given technology, the technical effectiveness utilized in accomplishing such projects can be determined.

These concepts of technology and its implementation can be incorporated into the structure of the expanded model of the software production loop. Productivity in the form of technical effectiveness is one of the two influencing factors on the rate of software production in that model. Figure 6-6 presents a causal loop diagram of the technology sector which produces this influence along with a measure of management effectiveness used throughout the other model sectors.

In the lower portion of Figure 6-6 the growth of technology is spurred by some gap between presently available technical effectiveness and some level of effectiveness required to complete a software project. This gap results in a pressure within the organization to improve the existing available technology which affects the advancement of the technological state of the art. The gap between actual available technical effectiveness and the technology required to complete the project is not precisely known to production management. Instead, that management develops a perception or estimate of that gap using its estimating capability which is a direct function of organizational management effectiveness. This management effectiveness also determines the translation of available technical effectiveness at the organization to that technical effectiveness utilized in software production.

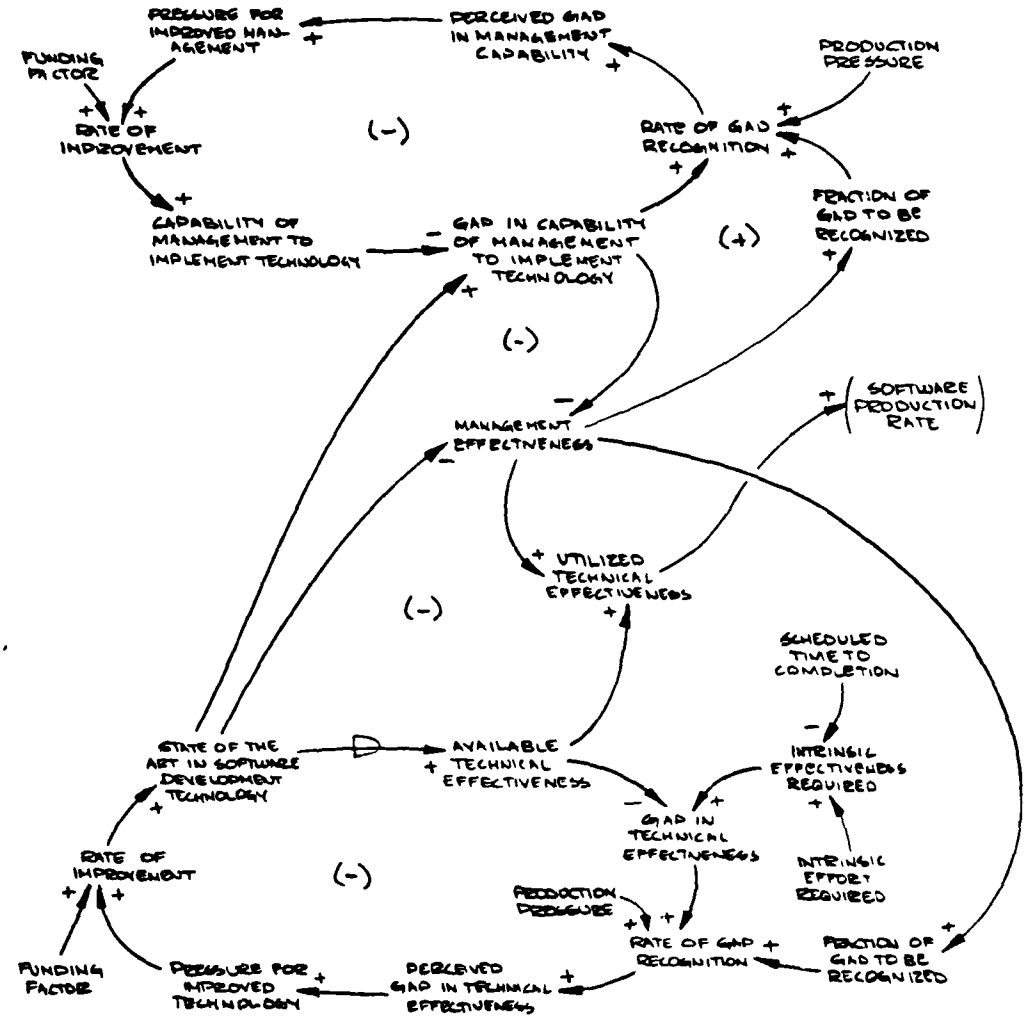


Fig. 6-6. Technology Sector

The difference between the technical state of the art and the technical effectiveness available at the organization is a time delay in transmitting information or knowledge to the organization's engineers. In a highly technical organization, continuing education, professional society membership, in-house technical libraries, and management attitude toward the same contribute to the determination of this delay.

The upper part of Figure 6-6 presents the growth of management capability to implement technology and its relation to management effectiveness as a feedback structure similar to the lower portion of the diagram. Again a gap, both actual and perceived by organization management, exists. This gap is the difference between the existing level of management capability to implement technology and the existing level of technology as shown in Figure 6-4. Again, the perception of that gap produces pressure for improvement to management capability, and a related rate of improvement.

The ultimate result of the feedback system presented in Figure 6-6 is utilized technical effectiveness as an influence on the software production rate.

The Manpower and Productivity Sector. The second influence on the rate of production of weapon system software is the manpower available. More specifically, this

rate is affected by the number and skill level of engineers working in the production and testing of weapon system software. In this paper the term engineer is used in its broadest sense to encompass all technical people contributing to the actual production and testing of weapon system software, such as programmers, systems analysts, software and hardware engineers, computer scientists, and others.

Currently, debate continues over the perception of a shortage of engineering manpower. The existence of a shortage and the factors contributing, such as engineering salaries and the rate of education of new engineers, are not fully defined. However, the fact that authorized levels of hiring and manning within government and the software industry are continually and dramatically unmet serves to substantiate the inflexibility of the supply of such manpower. For example, late in 1981, one of the Air Force's largest software support facilities had openings for 65 of 165 engineering authorizations (Ref 21). This limited supply of engineering manpower creates a continuing competition between acquisition, support and defense contractor organizations for the pool of available engineers. The feedback structure displaying this behavior is shown in the left side of Figure 6-7.

The total number of engineers involved in the production process can be translated into the number of

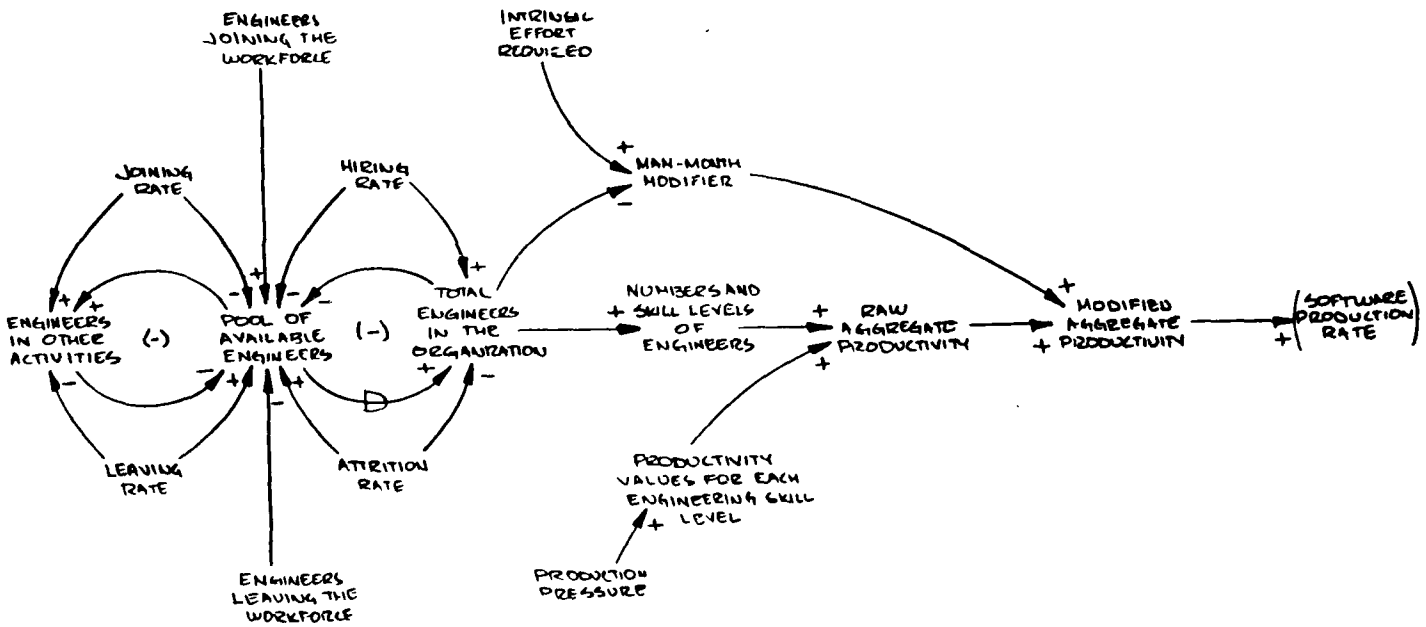


Fig. 6-7. Manpower and Productivity Sector

engineers classed into a variety of skill levels, as shown in the center of Figure 6-7. Engineers in training are a readily distinguishable classification as are experienced engineers. Some experienced engineers will be diverted from full time production duties to act as trainers for new engineers and as lower level managers. It is important to note such distinctions because the relative contribution of each engineering skill level to software production will vary.

Each group contributes to the overall hiring, transfer, and attrition rates within the organization. The hiring of new engineers does not immediately result in a change in manpower levels in the organization. There is usually a delay between the hiring and actual joining of the organization by an engineer. This delay is representative of a composite of delays. Newly educated engineers are usually recruited several months before graduation. Experienced engineers once recruited usually complete projects and provide notice to present employers before joining another organization.

In Chapter V, the influences on the software production rate of the generalized production loop were defined as being manpower and the technical effectiveness of tools and techniques employed. In system dynamics modelling, a rate can be interpreted as the volume of flow between two levels, or accumulations, in some unit time.

Thus, the software production rate in the production loop represents the transformation of work in progress to work accomplished, or the flow of some measure of work in a unit time between these two levels. In this case, the influence of manpower can be interpreted as that measure of the work accomplished per unit time; while the influence of technology can be interpreted as a multiplier of that measure, proportional to the level of technical effectiveness.

Although, numerous measures exist for quantifying work per unit time, the man-month per month measurement was selected for this study. This was the only measure that could commonly express amount of work inherent in the accomplishment of a variety of tasks, such as specifying system requirements, developing designs, writing lines of program code, and testing segments of coded software. Expressed simply, one man working full time on a project will contribute one man-month of work per month to the project, or will transform one man-month of work from in progress to accomplished per month.

In actuality, two factors will cause the contributions of individual engineers in different environments to vary from this one man-month per month constant. First, the employment of differing technologies with differing levels of effectiveness will cause actual contribution to vary from some theoretical limit possible with a totally effective technology. Second, the skill levels of the

engineers will cause actual contribution to vary from an assumed maximum for experienced engineers.

By multiplying the number of engineers in each skill level by some value relative to the productivity of an experienced engineer and then summing these products for all skill levels, one can obtain a raw measure of the amount of work accomplishable per month by all engineers in aggregate within the organization. In Figure 6-7 this value is represented by raw aggregate productivity. As shown in the diagram, the productivity value multipliers are variable in proportion to production pressure. This implies that within some range the ability of engineers in aggregate to accomplish work is sensitive to the influence of the production management.

A fallacious inference can be derived from using the man-month as a measure of work: that men and months are interchangeable commodities. Under this assumption of direct interchangeability the solution to the management problem of being behind schedule would be to assign a number of men to the task equivalent to the number of months behind schedule. The assumption of interchangeability of men and months is only valid when a task can be partitioned among many workers with no communications between them. Additionally, many tasks cannot be partitioned due to sequential constraints; the subtasks must be accomplished sequentially. In tasks that can be

partitioned but which require communication among the subtasks, the increasing effort of communication as workers are added must be added to the amount of work to be done. The best that can be accomplished is somewhat less than an even trade of men for months, as illustrated in Figure 6-8 (Ref 6:16-17).

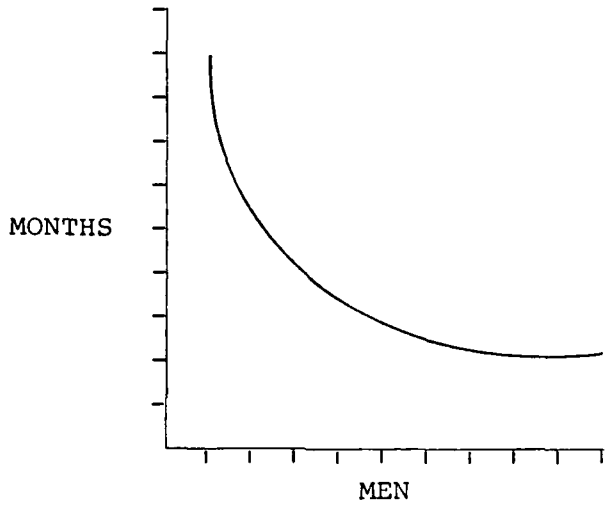


Fig. 6-8. Time Versus Number of Workers--Partitionable Task Requiring Communication

If each subtask must be separately coordinated with each other subtask, the effort increases as $n(n-1)/2$. Three workers require three times as much pairwise communication as two; four require six times as much as two. Moreover, if conferences between three, four, or more workers are required to jointly resolve matters; the added effort of communicating may fully counteract the benefits

of dividing the original task and produce the situation shown in Figure 6-9 (Ref 6:18-19).

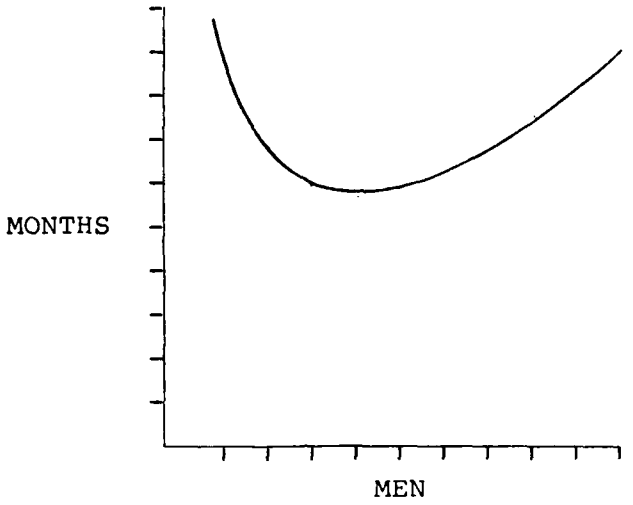


Fig. 6-9. Time Versus Number of Workers--Task with Complex Interrelaticnships

The point of this discussion is that the addition of engineers to the workforce of a software producing organization does not always directly increase the production of the organization, and may in fact decrease that production. In Figure 6-7, the man-month modifier adjusts the raw aggregate productivity of the organization's workforce by an amount related to the number of engineers assigned and the intrinsic effort required to complete the project. The effect is that on bigger projects the decline and reversal of marginal improvements in productivity as engineers are added is delayed.

The resulting modified aggregate productivity is the second influence, in conjunction with technical effectiveness on the software production rate of the generalized production loop.

Primary Influences on the Error and Rework Discovery Rates-- The Quality Management Sector

Both the error rate and the rework discovery rate of the software production process are expressions of software quality concern within the acquisition and support organizations. Similar to the partial dependence of the software production rate on the technical effectiveness of the organization, the error rate is dependent (inversely) on the effectiveness of organizational effort to correctly produce a software product. The discovery rate is dependent on the effectiveness of organization effort to detect production errors. Both efforts can be considered as concerns of software quality management.

Fisher and Light describe software quality as the "composite of the intrinsic attributes of computer software . . . which describe the degree of excellence of the . . . software [Ref 17:7]." A partial list of such attributes and their definitions is contained in Table 1. It may be noticed that the quality includes all aspects of the software: computer programs, data, and documentation. Also, an important point is that software quality is

TABLE 1

SOFTWARE QUALITY ATTRIBUTES [Ref 27:129]

Correctness	Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
Reliability	Extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources and code required by a program to perform a function.
Integrity	Extent to which access to software or data by unauthorized persons can be controlled.
Usability	Effort required to learn, operate, prepare input, and interpret output of a program.
Maintainability	Effort required to locate and fix an error in an operational program.
Testability	Effort required to test a program to insure it performs its intended function.
Flexibility	Effort required to modify an operational program.
Portability	Effort required to transfer a program from one hardware configuration and/or software system environment to another.
Reusability	Extent to which a program can be used in other applications--related to the packaging and scope of the functions that programs perform.
Interoperability	Effort required to couple one system with another.

introduced and enhanced in all phases of software acquisition and support.

Figure 6-10 presents a causal loop diagram of the quality management sector. The two major feedback loops in the diagram produce behavior similar to those in the technology sector. A discrepancy in the capability of the quality management system is observed as perceived and excess error rates. Improvement to the software quality management capability of the organization results from some pressure as a result of the perceived error situation. The software quality management capability is directed towards two efforts: effectiveness in testing for software quality and effectiveness in producing quality software. These two directions or effectiveness efforts directly influence the error and discovery rates of the software production process.

Peripheral influences on the behavior of the quality management sector and thus the behavior of the software production process itself include management policy concerning the funding of quality improvement efforts. Also, included is management policy concerning the allocation of quality management resources and effort between reducing the error rate by influencing quality in production, and improving the discovery of rework by influencing quality evaluation programs.

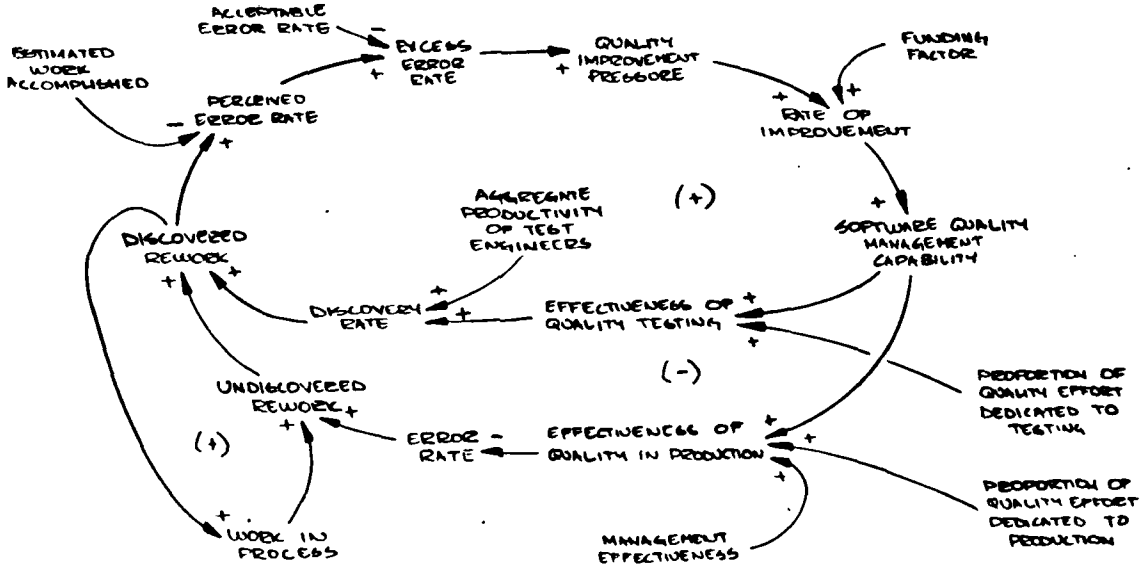


Fig. 6-10. Quality Management Sector

Chapter Summation

With the conclusion of this chapter the second stage of the system dynamics study of weapon system software acquisition and support is initially completed. I say initially complete to reinforce the repeatedly iterative aspect of the system dynamics methodology.

To this point, the causal components of the weapon system software acquisition and support system have been identified and their linkages have been established to present a conceptual model or theory of the system structure. As is characteristic of such models no effort has been made to force this structure into functional groupings. What may be interpreted as natural clumping or the existence of relatively closed subsystems has been exploited to enhance understanding of the model through the delineation of system and model sectors.

The recomposition of the sector models and the core software production process model result in a conceptual structural model of the expanded system presented in Figure 5-9. This overall system model illustrates a theory of the structure of the weapon system software acquisition and support system.

Although a generalized statement of the perceived system behavior was presented in earlier chapters, the remainder of this study initiates refinement of a theory of system behavior as a result of structure. The first

step, and the next presentation in this paper is the construction of a mathematical model of the system, implemented as a system dynamics flow diagram model and corresponding equations ready for computerization.

VII. A Mathematical Model of the Weapon System Software Acquisition and Support System

Introduction

The initial thesis of this investigation was that underlying the apparent difficulties in fulfilling expectations for weapon system cost and performance, there was a general lack of understanding of the complexities and interactions in the entire process of weapon system software acquisition and support. The foregoing presentation was developed to provide a conceptual insight into the system structure built of this complexity and interaction. The development of a mathematical model of that system, presented in this chapter, is intended to add precision and verification to, and hence increased understanding of, the previous descriptive and theoretical portions of this study. The solution to the problems of cost and performance may then result from the analysis and formulation of policy to guide system management based upon the behavior of the system as simulated by the computerization of this mathematical model.

This chapter will present a system dynamics flow diagram visualization of each of the five sectors presented in Chapters V and VI--production, production management,

technology, manpower and productivity, and quality management. Each sector will be modelled using the flow diagram symbology presented in Chapter III, corresponding equations, and description and illustration of model parameters. The flow diagram symbols each contain a variable name, symbol, and equation number. Following each diagram is a table of variable names, and associated abbreviations, dimensions, and equations describing it. The equation numbers are prefixed by a one or two character alpha code which corresponds to the sector names. For example, PM denotes production management and T denotes technology. Such equation numbers should aid the reader in associating equations in this chapter and Appendix A with the appropriate sector diagram. *Appendix A is a complete, documented* listing of the DYNAMO code for this model.

Production Sector

Probably the most simplistic sector model, the production sector flow diagram, is presented in Figure 7-1. The production sector model is essentially the same as the system dynamics model of the generalized software production process developed in Chapter V. Exceptions include the addition of informational linkages between the production sector and the other sectors, a model structure to initialize the level of work in progress, and a rate controlled by the production management sector to adjust the

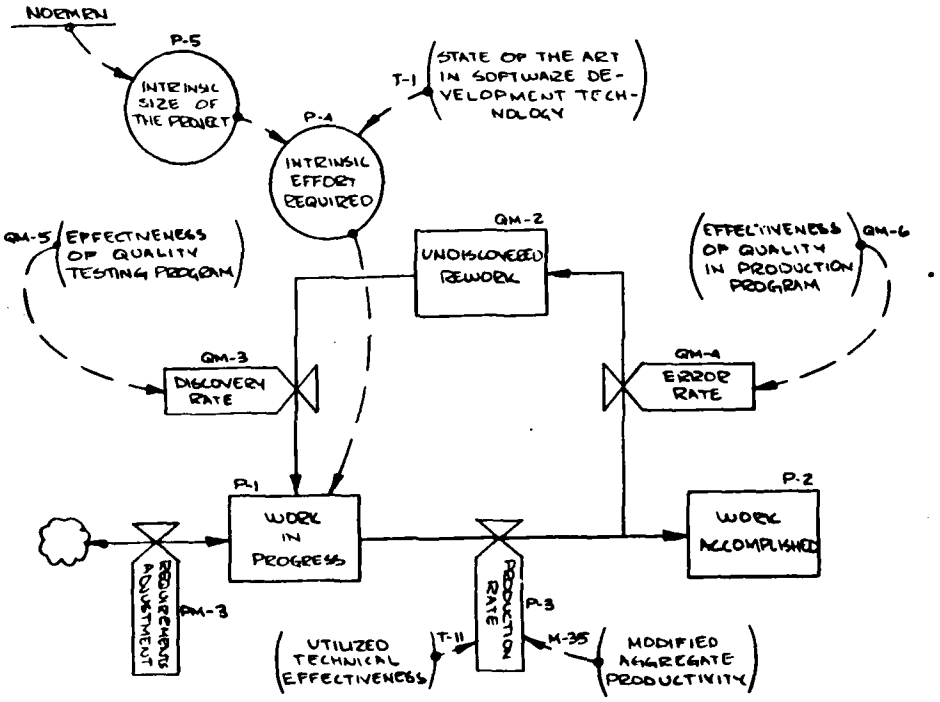


Fig. 7-1. Production Sector

TABLE 2

PRODUCTION SECTOR VARIABLES

Variable	Variable Name	Equation Describing	Dimension
WIP	Work in Progress	P-1	Man-Months of Effort
WAC	Work Accomplished	P-2	Man-Months of Effort
RPRO	Production Rate	P-3	Man-Months of Effort per Month
INEFRQ	Intrinsic Effort Required	P-4	Man-Months of Effort
INSZPJ	Intrinsic Size of the Project	P-5	(None)
TSOA	State of the Art in Software Technology	T-1	(None)
UTE	Utilized Technical Effectiveness	T-11	(None)
PEMAP	Modified Aggregate Productivity of Production Engineers	M-35	Man-Months of Effort per Month
RRQADJ	Rate of Adjustments to Requirements	PM-3	Man-Months of Effort per Month
RDISC	Rate of Discovery of Rework	QM-3	Man-Months of Effort per Month
RERR	Rate of Errors in Production	QM-4	Man-Months of Effort per Month

level of work in progress as software product requirements are varied in response to other system variables.

In Figure 7-1, the model is initialized by the generation of a normal random variate which determines the intrinsic size of each software project to be accomplished by the system. This intrinsic size compared to the level of the state of the art in software development or support technology results in a determination of the intrinsic effort required to complete the project. This intrinsic effort provides an initial value for the level of work in progress.

As stated in Chapter V the software production loop, and therefore the production sector, is principally modelled by three levels and three rates. The level of work in progress is reduced by the flow rate of software production to increase the level of work accomplished. The software production rate is determined by the product of the auxillary variables, utilized technical effectiveness from the technology sector and modified aggregate productivity from the manpower and productivity sector.

The feedback of errors in work accomplished to undiscovered rework to discovered rework adding to work in progress is modelled indirectly in the quality management sector. This loop is shown explicitly in the production sector flow diagram, Figure 7-1.

Table 2 lists the variable names, abbreviations, dimensions, and associated equation numbers for the production sector model. The sector equations are:

```
WIP.K=WIP.J+DT(RDISC.JK-RRQAOJ.JK-RPRO.JK)      P-1,Level
WIP=INEFRQ                                      P-1N,Initial
WAC.K=WAC.J+DT(RPRO.JK-RERR.JK)                 P-2,Level
WAC=0                                           P-2N,Initial
RPRO.KL=(UTE.K)(PEMAP.K)                        P-3,Rate
INEFRQ=INSZPJ/TSOA                              P-4N,Initial
INSZPJ=NORMN(MEAN,STANDARD DEVIATION)           P-5N,Initial
```

In equation P-5N the intrinsic size of each project undertaken by the process is pseudorandomly generated from a normal distribution described by the model user with its mean and standard deviation.

Production Management Sector

The production management sector is the first of several sectors to be described which are dominated by the flow of information used to determine the result of decision-making processes represented by rates. The flow diagram for this sector model is shown in Figure 7-2. Table 3 lists the variables used in the production sector model.

In Figure 7-2 the information network shown as auxillary variables and information flows circulates around the estimation of variables involved in the transformation of work in progress to work accomplished. Estimated work remaining is production management's estimated parallel to the actuality of work in progress. This level is

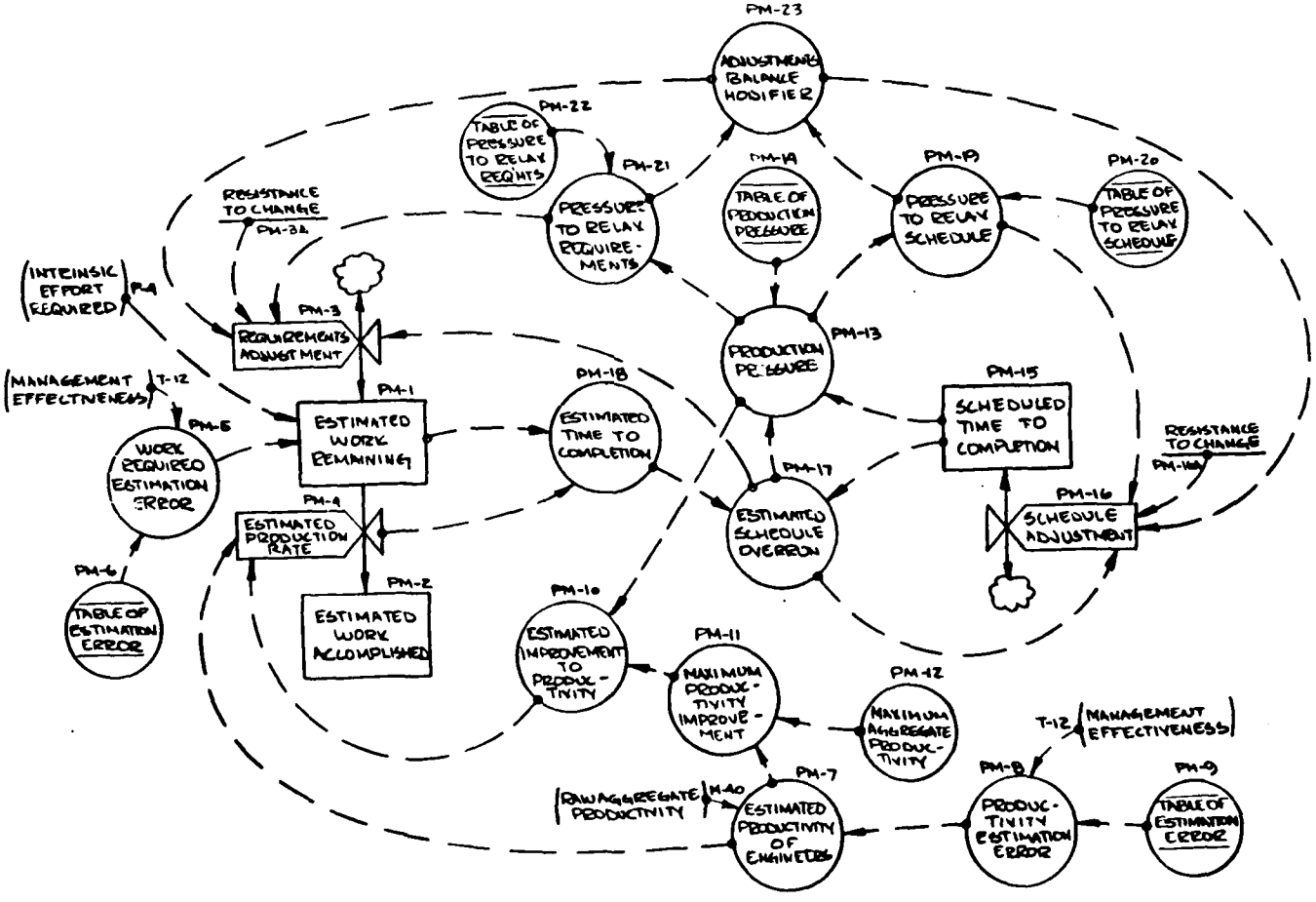


Fig. 7-2. Production Management Sector

TABLE 3

PRODUCTION MANAGEMENT SECTOR VARIABLES

Variable	Variable Name	Equation Describing	Dimension
EWR	Estimated Work Remaining	PM-1	Man-Months of Effort
EWA	Estimated Work Accomplished	PM-2	Man-Months of Effort
RRQADJ	Rate of Adjustments to Requirements	PM-3	Man-Months of Effort per Month
RCHGR	Resistance to Changes in Requirements	PM-3A	(None)
REPRO	Estimated Rate of Production	PM-4	Man-Months of Effort per Month
WEER	Error in Estimate of Work Required	PM-5	(None)
TWEER	Table of WEER (see Figure 7-3)	PM-6	(None)
EPE	Estimated Productivity of Production Engineers	PM-7	Man-Months of Effort per Month
EIPR	Estimated Improvement to the Production Rate	PM-10	Man-Months of Effort per Month
MXIMP	Maximum Productivity Improvement	PM-11	(None)
MAXPR	Maximum Aggregate Productivity of Production Engineers	PM-12	Man-Months of Effort per Month
PRP	Production Pressure	PM-13	(None)
STTC	Scheduled Time to Completion	PM-15	Months
RSCHADJ	Rate of Adjustments to Schedule	PM-16	Months per Month

TABLE 3--Continued

Variable	Variable Name	Equation Describing	Dimension
RCHGS	Resistance to Changes in Schedule	PM-16A	(None)
ESCHO	Estimated Schedule Overrun	PM-17	Months
ETTC	Estimated Time-to-Completion	PM-18	Months
RRSCH	Pressure to Relax Schedule	PM-19	(None)
PRREQ	Pressure to Relax Requirements	PM-21	(None)
ABM	Adjustments Balancing Modifier	PM-23	(None)
INEFRQ	Intrinsic Effort Required	P-4	Man-Months of Effort
MGTEF	Management Effectiveness	T-12	(None)

initially set by management's estimate of the intrinsic effort required to complete a project. This estimate is imprecise by the value of a work required estimation error which is inversely related to management effectiveness as shown in Figure 7-3.

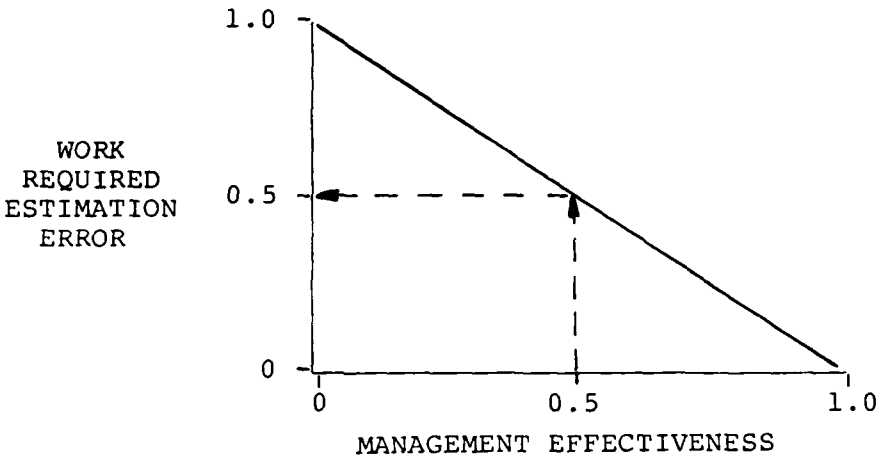


Fig. 7-3. Table of Work Required Estimation Error versus Management Effectiveness

As time advances, production management's estimate of the software production rate provides a decrease in the level of estimated work remaining, and a corresponding increase in the level of estimated work accomplished. The level of estimated work remaining is also influenced by the rate of adjustment of software production requirements in response to other system variables. The equations listed below describe these relationships:

```
EWR.K=EWR.J+DT(RRQADJ.JK-REPRO.JK)          PM-1,Level
EWR=INEFRQ(1-WEER)                          PM-1,Initial
EWA.K=EWA.J+DT(REPRO.JK)                    PM-2,Level
EWA=0                                       PM-2N,Initial
```

Whereas, the actual software production rate is the product of aggregate productivity and technical effectiveness; the estimated production rate is the sum of management's estimate of the aggregate productivity of production engineers and its estimate of how much that productivity can be increased as a result of pressure for increased production. The estimated productivity of production engineers results from management's evaluation of raw aggregate productivity as perturbed in error by a productivity estimation error inversely related to management effectiveness. This relation is identical to that shown for work required estimation error in Figure 7-3. Raw aggregate productivity is essentially the product of the numbers and skill levels of engineers times some estimates of productivity for each skill level. Engineer productivity is expressed as man-months of effort per month.

The estimated improvement to productivity as a result of production pressure is bounded by some maximum improvement. This maximum improvement is variable depending upon the relation of maximum possible productivity to the estimated productivity of production engineers. These relationships are represented in the following equations:

```
REPRO.KL=(EPE.K)(1+EIPR.K)                          PM-4,Rate
EPE.K=(PERAP.K)(1+PEER.K)                           PM-7,Auxillary
EIPR.K=(MXIMP.K)(PRP.K)                             PM-10,Auxillary
MXIMP.K=(MAXPR.K/EPE.K)-1                           PM-11,Auxillary
MAXPR.K=(PCEIT.K)(PEIT.K)+(PCEXE.K)
        (PEXE.K)+(PCMTE.K)(PMTE.K)
```

The pressure for increased production which determines management's perception of how much improvement can be made to productivity is the result of production management's comparison of the scheduled time to completion to the magnitude of any estimated schedule overrun. This relationship is shown in Figure 7-4. The concept captured in this figure is that management's early treatment of an estimated schedule overrun is not required, that time is still available to catch up. However, as scheduled completion approaches and the estimated overrun looms as large as ever, then pressure is greatly increased to increase production.

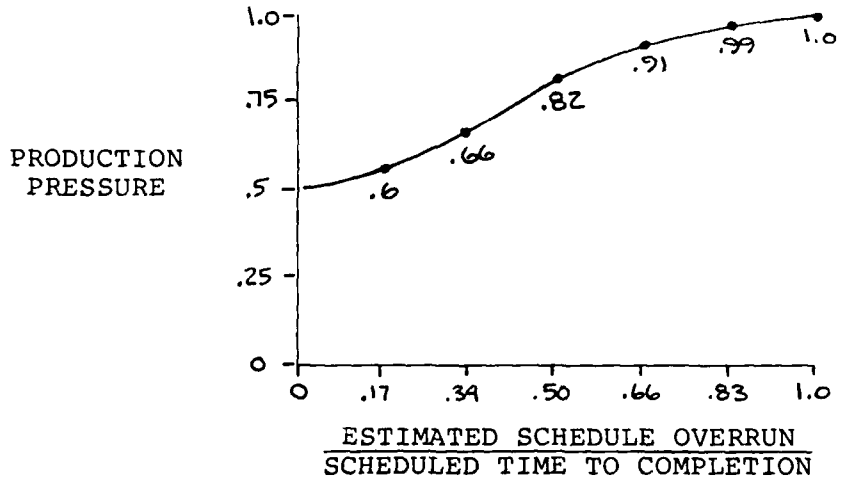


Fig. 7-4. Table of Production Pressure versus a Comparison of Overrun to Schedule

Estimated schedule overrun results from production management's comparison of scheduled and estimated time to completion. As a project begins both times are equal,

because the schedule is derived from management's original estimate of effort required to complete the project. As time advances, scheduled time to completion is correspondingly reduced, but is also adjusted as a result of decisions to extend or constrain the schedule in response to other system variables, as reflected in the rate of schedule adjustment. Management computes the estimated time to completion by dividing its estimate of work remaining by its estimate of the production rate. These relationships are described by the equations following:

```
STTC.K=STTC.J+DT(RSCHADJ.JK-1)          PM-15,Level
STTC=ETTC                               PM-15N,Initial
ESCHO.K=MAX((ETTC.K-STTC.K),0)          PM-17,Auxillary
ESCHO=0                                 PM-17N,Initial
ETTC.K=EWR.K/REPRO.JK                   PM-18,Auxillary
```

In the development of complex, technological products, such as weapon system software, a well-known triad exists, which includes cost, schedule, and system performance. Management of the production process involves continual tradeoff of these parameters against one another. Because this study assumes that a project undertaken by the acquisition and support system is funded and will continue to be funded at a reasonable level, matters of cost and budget are not considered. However, schedule and performance are considered as potential tradeoffs and are affected by production pressure in the production management sector model.

Production pressure directly results in both pressure to relax performance requirements and to relax or extend the project schedule, as shown for both cases in Figure 7-5. The use of an adjustments balancing modifier in the sector model insures that both adjustments are proportional to a total potential adjustment.

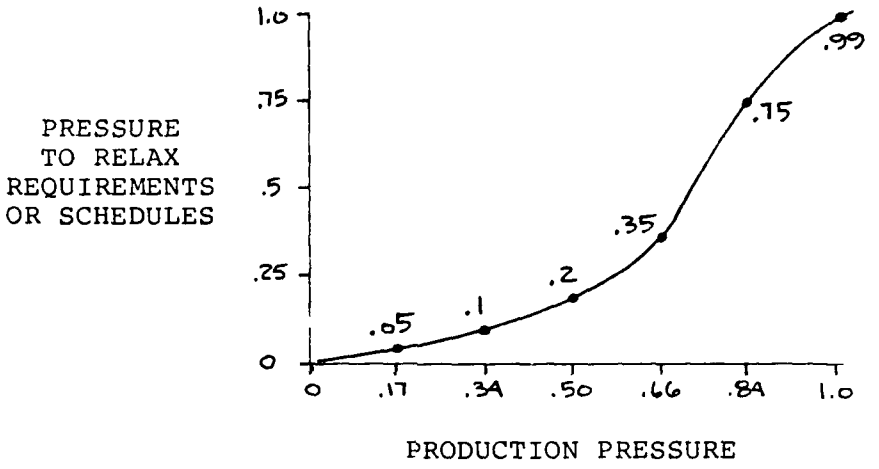


Fig. 7-5. Table of Pressure to Relax Requirements or Schedule versus Production Pressure

Both pressure to relax the schedule and pressure to relax requirements are countered by an organizational resistance to change once a project is initiated. In the production management sector model this resistance was characterized as a constant value, in order to represent a threshold above which lower level management's pressure for change would have to advance before any change in requirements or schedule would be allowed.

These equations describing these relationships and closing the feedback system of Figure 7-2 are listed below:

```
RRQADJ.KL=(PRREQ.K-RCHGR)(ESCHO.K)          PM-3,Rate
          (RREQ.K/ABM.K)
RCHGR=.2                                    PM-3A,Constant
RSCHADJ.KL=(PRSCH.K-RCHGS)(ESCHO.K)         PM-16,Rate
           (PRSCH.K/ABM.K)
RCHGS=.2                                    PM-16A,Constant
ABM.K=PRREQ.K+PRSCH.K                       PM-23,Auxillary
```

Technology Sector

Within the technology sector, the measures of effectiveness of the tools and techniques used in the software production process are generated. Technical and management effectiveness are developed within separate but interfacing decision structures, as shown in Figure 7-6.

Technology as discussed in Chapter VI, can be modelled as an increasing level of the state of the technological art. This rate of growth of this level is determined by a rate of improvement resulting from the product of organizational pressure for improved technology and the percent of desired improvement actually to be funded. The pressure for improved technology results from the perception of management and engineers of a gap or shortfall in the technical effectiveness of the organization needed to successfully complete a software project. Figure 7-7 shows this increase in pressure for improved technology as the perceived gap in technical effectiveness increases.

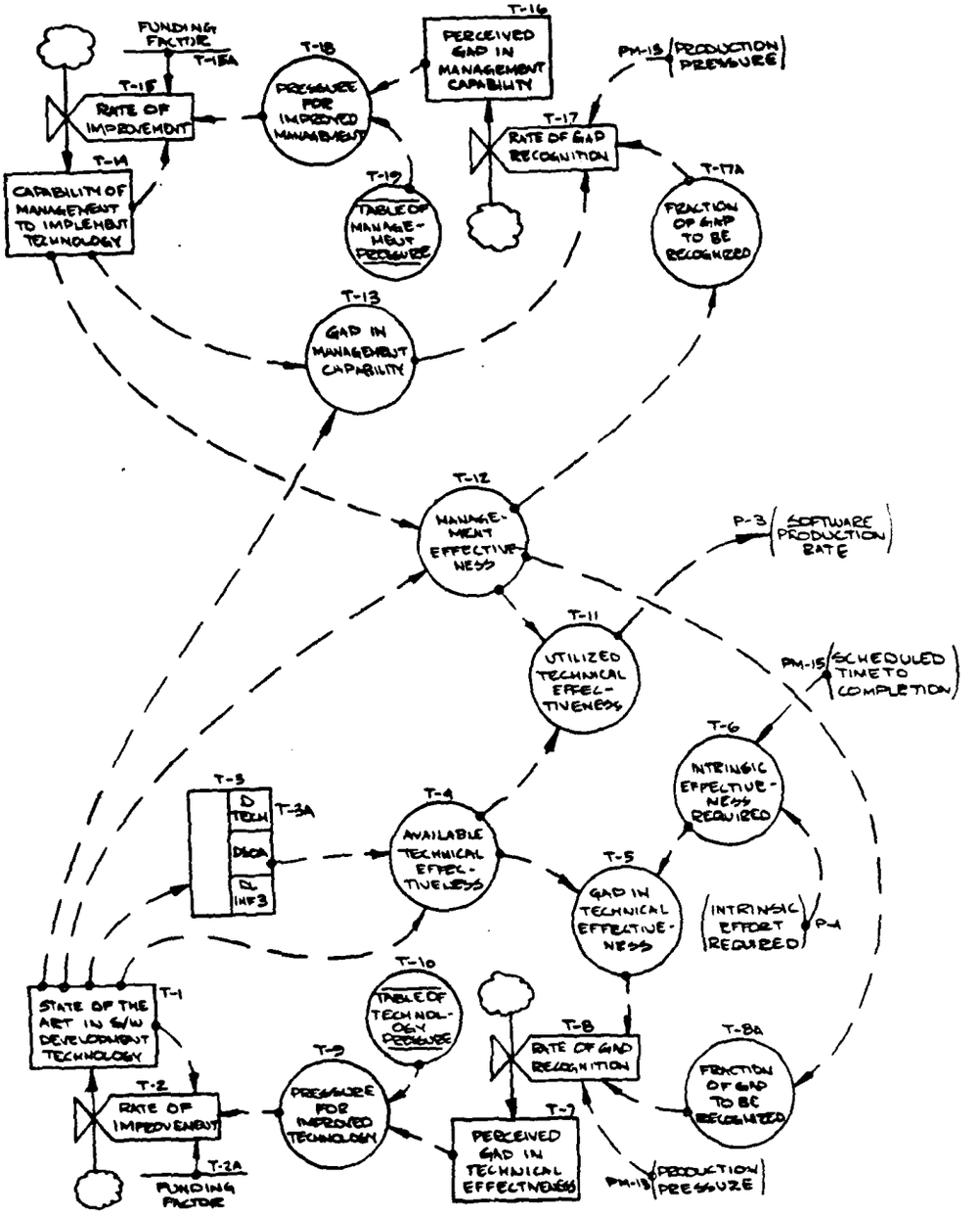


Fig. 7-6. Technology Sector

TABLE 4

TECHNOLOGY SECTOR VARIABLES

Variable	Variable Name	Equation Describing	Dimension
TSOA	State of the Art in Software Technology	T-1	(None)
RTIMP	Rate of Improvement in Technology State of the Art	T-2	(None)
TIFF	Technology Improvement Funding Factor	T-2A	(None)
DSOA	Delayed State of the Art Technology	T-3	(None)
DTECH	Delay in Acquiring State of the Art Technology	T-3A	Months
ATE	Available Technical Effectiveness	T-4	(None)
TGAP	Gap in Technical Effectiveness	T-5	(None)
IEFRQ	Intrinsic Effectiveness Required	T-6	(None)
PTGAP	Perceived Gap in Technical Effectiveness	T-7	(None)
RTGPR	Rate of Recognition of Technical Gap	T-8	(None)
TGAPFR	Fraction of Gap Recognized	T-8A	(None)
TPRES	Pressure for Improved Technology	T-9	(None)
UTE	Utilized Technical Effectiveness	T-11	(None)
MGTEF	Management Effectiveness	T-12	(None)

TABLE 4--Continued

Variable	Variable Name	Equation Describing	Dimension
MCIT	Management Capability to Implement Technology	T-13	(None)
RMIMP	Rate of Improvement in Management Capability	T-14	(None)
MIFF	Management Capability Improvement Funding Factor	T-14A	(None)
PMGAP	Perceived Gap in Management Capability	T-15	(None)
RMGPR	Rate of Recognition of a Management Capability Gap	T-16	(None)
MGAPFR	Fraction of Management Gap Recognized	T-16A	(None)
MGAP	Gap in Management Capability to Implement Technology	T-17	(None)
MPRES	Pressure for Improved Management	T-19	(None)
INEFRQ	Intrinsic Effort Required	P-4	Man-Months of Effort

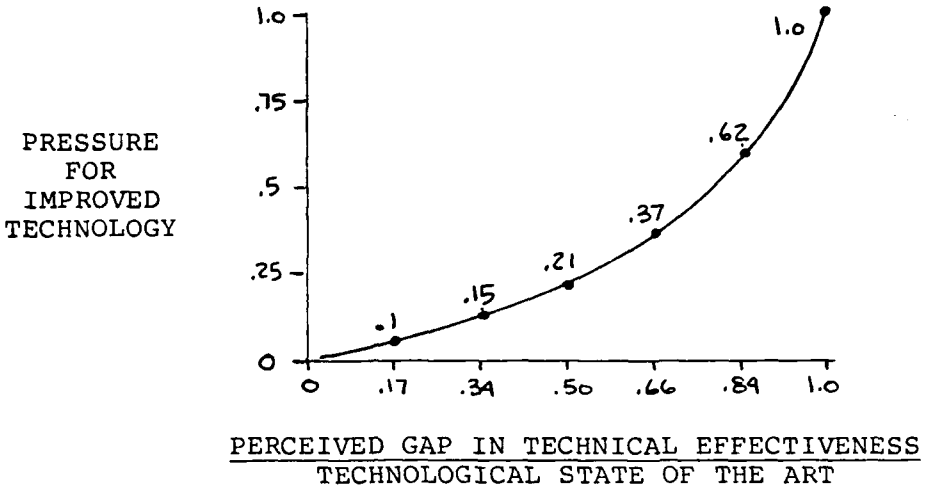


Fig. 7-7. Table of Pressure for Improved Technology versus a Comparison of a Perceived Gap to the State of the Art

The concept presented is the accelerating pressure as the size of the perceived gap increases in relation to the level of technology.

The perceived gap itself is the organization's estimate of an actual shortfall in technical effectiveness. The precision of this estimate or perception is directly related to the effectiveness of organizational management to understand and implement the technology available. Production pressure by increasing the air of crisis artificially heightens the accuracy of the perception.

The actual gap in technical effectiveness is established initially by comparing the intrinsic effectiveness required to complete a project to the general level of technical effectiveness available in the organization

before a project is begun. This intrinsic effectiveness is provided by comparing the first scheduled time to completion to the intrinsic effort required to do the job.

The last factor in closing the loop which produces continual effort for the improvement of technology is the delay in transmitting the latest software technology to the organization to become available technical effectiveness. This delay is described in the sector model as a constant time for information in the form of professional journals, technology reports, continuing education, and seminars among other media, to provide the latest technology information to the production engineer. A comparison of this delayed level of technological information to the level of the technological state of the art provides the available technical effectiveness at the organization.

The relationships presented above are described in the following equations:

```
TSOA.K=TSOA.J+DT(RTIMP.JK)                    T-1,Level
RTIMP.KL=(TPRES.K)(TIFF)(TSOA.K)              T-2,Rate
TIFF=.75                                      T-2A,Constant
DSOA.K=DLINF3(TSOA.K,DTECH)                   T-3,Auxillary
DTECH=6                                       T-3A,Constant
ATE.K=DSOA.K/TSOA.K                           T-4,Auxillary
TGAP.K=MAX((IEFRQ.K-ATE.K),0)                 T-5,Auxillary
IEFRQ.K=INEFRQ.K/STTC.K                       T-6,Auxillary
PTGAP.K=PTGAP.J+DT(RTGPR.JK)                  T-7,Level
RTGPR.KL=(TGAP.K)(TGAPFR.K)(PRP.K+1)          T-8,Rate
TGAPFR.K=MGTEF.K                              T-8A,Auxillary
```

Similar to the design structure supporting improvements to technology, a feedback structure exists to support improvements in the organization's capability to manage

the implementation of technology. The concept of a gap in management capability, as presented in Chapter V, is implemented in the technology sector model by comparing the level of technology to the level of management capability to implement that technology. As with technical effectiveness, this gap is preceived in relation to management effectiveness and production pressure, and its perception results in pressure to improve management capability. That pressure accelerates as the comparison of the perceived gap to the technological state of the art increases, similarly to the relationship in Figure 7-7.

The relationships inherent in the decision structure supporting the improvement of management capability are described in the following equations:

```
MGTEF.K=MCIT.K/TSOA.K                          T-12,Auxillary
MCIT.K=MCIT.J+DT(RMIMP.K)                      T-13,Level
MCIT=TSOA                                      T-13N,Initial
RMIMP.KL=(MPRES.K)(MIFF)(MCIT.K)               T-14,Rate
MIFF=.75                                       T-14A,Constant
PMGAP.K=PMGAP.J+DT(RMGPR.JK)                   T-15,Level
RMGPR.KL=(MGAP.K)(MGAPFR.K)(PRP.K+1)           T-16,Rate
MGAPFR.K=MGTEF.K                               T-16A,Auxillary
MGAP.K=TSOA.K-MCIT.K                           T-17,Auxillary
```

The major explicit effect of the technology sector is upon the software production rate. The product of available technical effectiveness and management effectiveness results in a utilized technical effectiveness directly affecting the production rate. This relationship is described by the following equation:

```
UTE.K=(ATE.K)(MGTEF.K)                         T-11,Auxillary
```

Manpower and Productivity Sector

The manpower and productivity sector can be described as consisting of a relatively closed flow of engineering manpower and its integrating information network, and a second information network of auxillary variables and information flows resulting in aggregate productivity measures for production and test engineers.

The flow of engineering manpower is dominated by two loops into and out of a large pool of available engineers. This structure is shown in Figure 7-8. Engineers flow from the pool of available engineers to the engineering workforce of the weapon system software organization as controlled by the hiring rate of that organization. A composite loss rate across all skill levels in the organization determines the return of engineers to the available pool.

Outside of the organization, other engineers are involved in engineering activities with other organizations which remove them from the pool of available engineers. However, a continuing portion of these engineers rejoin the pool while a portion of available engineers joins these other activities.

The pool of available engineers is also influenced from outside the system by the education and training of new engineers who join the pool, and the attrition of established engineers who retire or embark upon new careers.

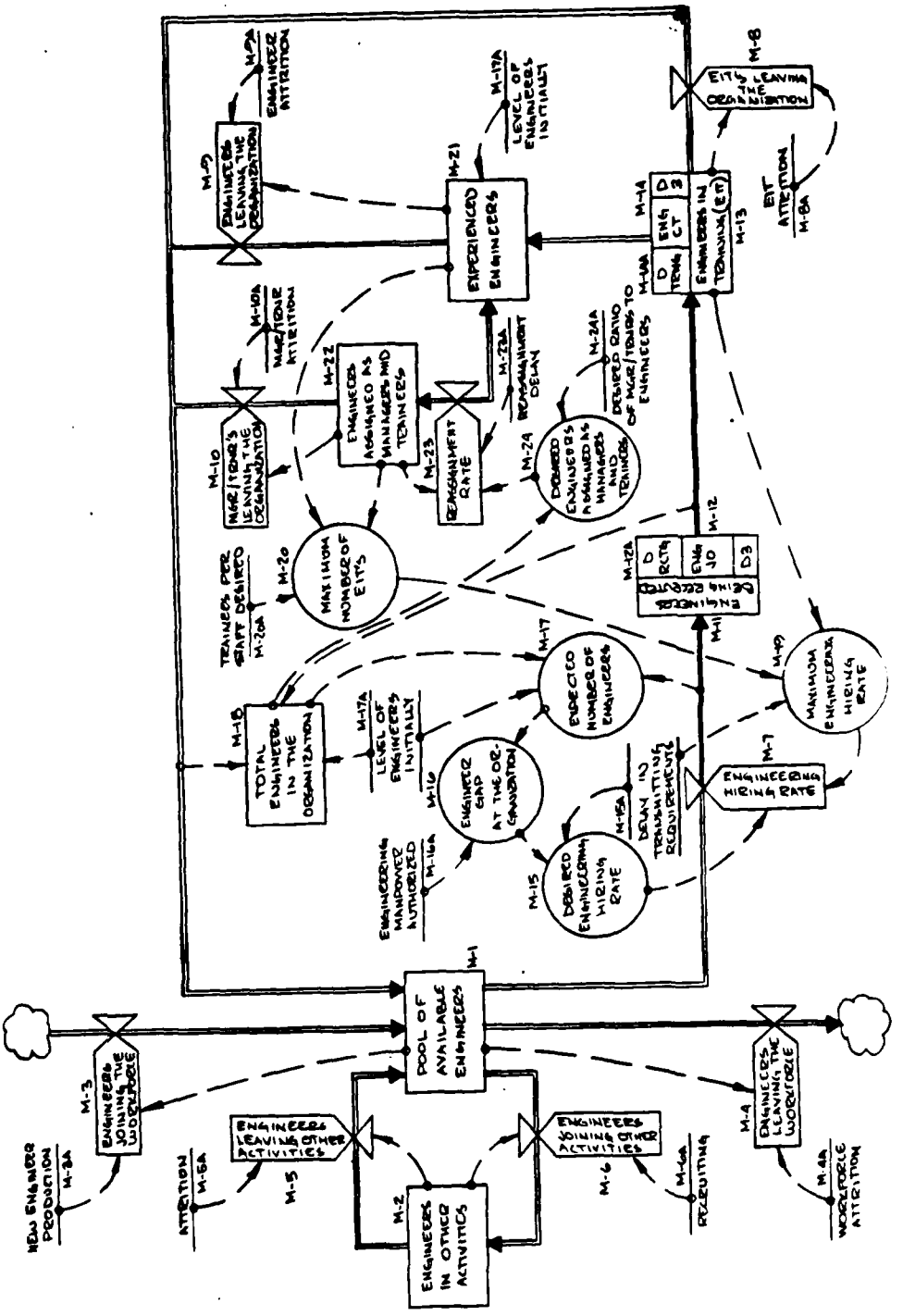


Fig. 7-8. Manpower and Productivity Sector (Part 1)

TABLE 5

MANPOWER AND PRODUCTIVITY SECTOR
VARIABLES (PART 1)

Variable	Variable Name	Equation Describing	Dimension
ENGPA	Pool of Available Engineers	M-1	Engineers
ENGOA	Engineers in Other Activities	M-2	Engineers
REJW	Rate Engineers Join the Workforce	M-3	Engineers per Month
RELW	Rate Engineers Leave the Workforce	M-4	Engineers per Month
RELOA	Rate Engineers Leave Other Organizations	M-5	Engineers per Month
REJOA	Rate Engineers Join Other Organizations	M-6	Engineers per Month
FNEP	New Engineer Production Factor	M-3A	(None)
FWFA	Workforce Attrition Factor	M-4A	(None)
FOAA	Attrition in Other Organizations Factor	M-5A	(None)
FOAR	Recruiting by Other Organizations Factor	M-6A	(None)
RHIRE	Rate of Engineer Hiring	M-7	Engineers per Month
REITL	Rate Engineers in Training Leave the Organization	M-8	Engineers per Month
FEITA	Engineer in Training Attrition Factor	M-8A	(None)

TABLE 5--Continued

Variable	Variable Name	Equation Describing	Dimension
REXEL	Rate Experienced Engineers Leave the Organization	M-9	Engineers per Month
FEXEA	Experienced Engineer Attrition Factor	M-9A	(None)
RMTEL	Rate Engineer Managers/ Trainers Leave the Organization	M-10	Engineers per Month
FMTEA	Engineer Manager/Trainer Attrition Factor	M-10A	(None)
ENGBR	Engineers Being Recruited	M-11	Engineers
ENGJO	Rate Recruited Engineers Join the Organization	M-12	Engineers per Month
DRCTG	Delay in Recruiting Engineers	M-12A	Months
ENGIT	Engineers in Training	M-13	Engineers
ENGCT	Rate Engineers Complete Training	M-14	Engineers per Month
DTRNG	Delay in Training Engineers	M-14A	Months
DEHR	Desired Engineer Hiring Rate	M-15	Engineers per Month
ENGAP	Engineer Gap at the Organization	M-16	Engineers
ENGMA	Engineering Manpower Authorized	M-16A	Engineers
ENGEN	Expected Number of Engineers	M-17	Engineers
LEI	Level of Engineers Initially	M-17A	Engineers

TABLE 5--Continued

Variable	Variable Name	Equation Describing	Dimension
ENGTOT	Total Engineers in the Organization	M-18	Engineers
MEHR	Maximum Engineer Hiring Rate	M-19	Engineers
MNEIT	Maximum Number of Engineers in Training	M-20	Engineers
TPSD	Trainees per Staff Desired	M-20A	Engineers
ENGEX	Experienced Engineers	M-21	Engineers
ENGMT	Engineer Managers/Trainees	M-22	Engineers
RREA	Rate of Reassignment of Engineers as Managers/ Trainers	M-23	Engineers per Month
DREA	Delay in Reassigning Engineers as Managers/ Trainers	M-23A	Months
DEAMT	Desired Engineers Assigned as Managers/ Trainers	M-24	Engineers
FDRMT	Desired Ratio of Managers/Trainers to Total Engineers	M-24A	(None)

The equations describing the immediate flow of manpower into and out of the pool of available engineers are listed following:

```
ENGPA.K=ENGPA.J+DT(REJW.JK+RELOA.JK+          M-1,Level
        REITL.JK+REXEL.JK+RMTEL.JK-
        RELW.JK-REJOA.JK-RHIRE.JK)
ENGOA.K=ENGOA.J+DT(REJOA.JK-RELOA.JK)         M-2,Level
REJW.KL=(FNEP)(ENGPA.K)                       M-3,Rate
FNEP=.019                                     M-3A,Constant
RELW.KL=(FWFA)(ENGPA.K)                       M-4,Rate
FWFA=.019                                     M-4A,Constant
RELOA.KL=(FOAA)(ENGOA.K)                      M-5,Rate
REJOA.KL=(FOAR)(ENGOA.K)                      M-6,Rate
FOAR=.021                                     M-6A,Constant
REITL.KL=(FEITA)(ENGIT.K)                     M-8,Rate
FEITA=.01                                     M-8A,Constant
REXEL.KL=(FEXEA)(ENGEX.K)                     M-9,Rate
FEXEA=.025                                    M-9A,Constant
RMTEL.KL=(FMTEA)(ENGMT.K)                     M-10,Rate
FMTEA=.01                                     M-10A,Constant
```

The acquisition of engineering manpower and its movement through organizational skill levels and out of the organization is modelled by the remainder of Figure 7-8. The engineering hiring rate by the organization from the pool of available engineers is determined from the minimum of the desired engineering hiring rate or a maximum supportable engineering hiring rate.

The desired engineering hiring rate is the positive difference of the level of engineering manpower authorized and the expected number of engineers at the organization, delayed by the time in translating this requirement to a decision to hire a new engineer. That positive difference is the engineer gap at the organization, and the expected number of engineers at the

organization is the sum of the total engineers at the organization and any expected new hirings.

The maximum engineering hiring rate is the positive difference of the level of engineers in training at the organization and the maximum number of trainees supportable based upon a desired trainee to staff ratio; that difference also delayed by time in translation to a decision to hire a new engineer.

As discussed in Chapter VI, engineers newly hired do not immediately arrive at the organization as part of the workforce but are delayed in recruiting for some time to complete previously incurred commitments before moving to the weapon system software organization. These same engineers are initially maintained in the organization as engineers in training, who either leave the organization before completing training or progress to become experienced engineers. Rather than being characterized as a formal training process, the delay as an engineer in training is more accurately described as a period in which the newly hired engineer, whether fresh from college or as an experienced engineer hired from another organization, becomes familiar with the way his new working environment is structured and operates. This enables him to develop and progress to an effective method of operating in that environment.

The equations describing the movement of engineering manpower from outside the organization to effective operation within it are listed below:

```
RHIRE.KL=MIN(DEHR.K,MEHR.K)                          M-7,Rate
ENGBR.K=ENGBR.J+DT(RHIRE.JK-ENGJO.JK)                M-11,Level
ENGJO.KL=DELAY3(RHIRE.JK,DRCTG)                      M-12,Rate
DRCTG=3                                              M-12A,Constant
ENGIT.K=ENGIT.J+DT(ENGJO.JK-
        ENGCT.JK-REITL.JK)                           M-13,Level
ENGCT.KL=DELAY3(ENGJO.JK,DTRNG)                      M-14,Rate
DTRNG=6                                              M-14A,Constant
DEHR.K=ENGAP.K/DTREQ                                 M-15,Auxillary
DTREQ=1.5                                            M-15A,Constant
ENGAP.K=MAX((ENGMA-ENGEN.K),0))                      M-16,Auxillary
ENGEN.K=ENGTOT.K+DT(RHIRE.JK)                        M-17,Auxillary
ENGTOT.K=ENGTOT.J+DT(ENGJO.JK-
         REITL.JK-REXEL.JK-RMTEL.JK)                 M-18,Level
MEHR.K=MAX((MNEIT.K-ENGIT.K)/DTREQ,0)                M-19,Auxillary
MNEIT.K=(TSPD)(ENGMT+ENGEX.K)                        M-20,Auxillary
TSPD=.3                                              M-20A,Constant
```

Experienced engineers serve within the organization in two roles. First, they can work full-time as production or test engineers directing their full productivity to the software product. Second, they may serve as trainers for new engineers or as low level managers of other engineers. In the second case something less than their full productivity will be directed towards weapons system software production. The flow of experienced engineers between levels representing each of the two cases is determined by a reassignment rate. This rate is the result of a desired ratio of total engineers assigned to each of the levels and a delay in making the reassignment.

By maintaining a record of engineers initially assigned to the orgnization and all hirings and losses

over time, a level of total engineers in the organization can be accounted. This level is used variously within the equations for this sector model. The total number of engineers in the organization, the expected number of engineers, and the level of experienced engineers are all set equal to the level of engineers initially when the simulation is begun.

The last variables closing the loop of manpower flow within the organization are the rates at which experienced engineers and manager/trainers leave the organization. Both of these rates are determined by an attrition factor for each engineering category.

The remaining equations for this portion of the manpower and productivity sector are listed below:

```
ENGEX.K=ENGEX.J+DT(ENGCT.JK-REXEL.JK
        -RREA.JK)                          M-21,Level
ENGEN=LEI                                  M-17N,Initial
ENGTOT=LEI                                 M-18N,Initial
ENGEX=LEI                                  M-21N,Initial
LEI=100                                    M-17A,Constant
ENGMT.K=ENGMT.J+DT(RREA.JK-RMTEL.JK)       M-22,Level
ENGMT=0                                    M-22N,Initial
RREA.KL=(DEAMT.K-ENGMT.K)/DREA             M-23,Auxillary
DREA=1                                     M-23A,Constant
DEAMT.K=(FDRMT)(ENGTOT.K)                  M-24,Auxillary
FDRMT=.2                                   M-24A,Constant
```

A second information network of auxillary variables and information flows results in aggregate productivity measures for production and test engineers. The flow diagram model of this portion of the manpower and productivity sector is shown in Figure 7-9.

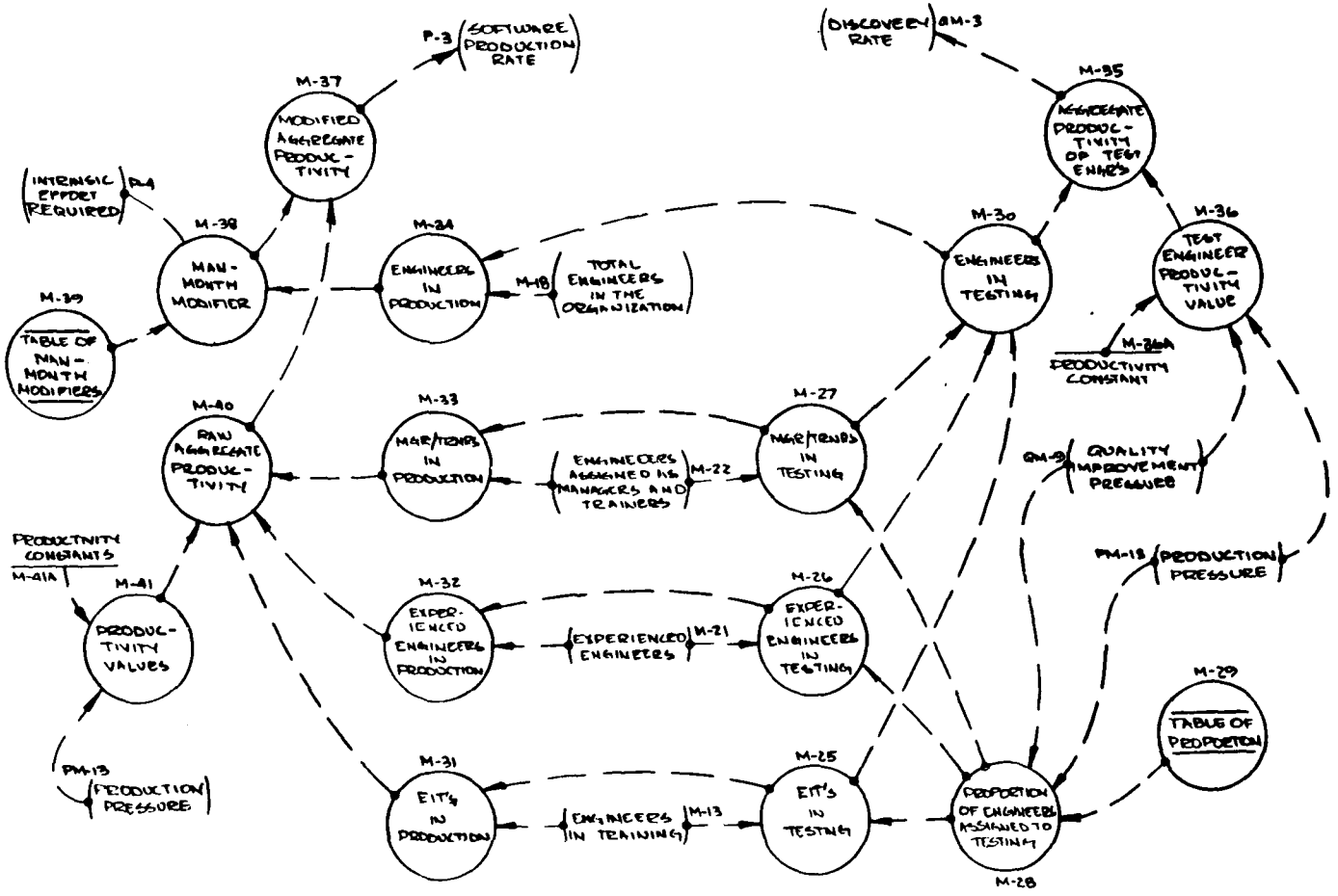


Fig. 7-9. Manpower and Productivity Sector (Part 2)

TABLE 6

MANPOWER AND PRODUCTIVITY SECTOR
VARIABLES (PART 2)

Variable	Variable Name	Equation Describing	Dimension
TEIT	Test Engineers in Training	M-25	Engineers
TEXE	Experienced Engineers in Testing	M-26	Engineers
TMTE	Engineer Managers/ Trainers in Testing	M-27	Engineers
PEAT	Proportion of Engineers Assigned to Testing	M-28	(None)
TPEAT	Table of Proportion of Engineers Assigned to Testing Versus Quality and Production Pressures	M-29	(None)
TENGR	Engineers in Testing	M-30	Engineers
PEIT	Production Engineers in Training	M-31	Engineers
PEXE	Experienced Engineers in Production	M-32	Engineers
PMTE	Engineer Managers/ Trainers in Production	M-33	Engineers
PENGR	Engineers in Production	M-34	Engineers
TEAP	Aggregate Productivity of Test Engineers	M-35	Man-Months for Effort per Month
TEPV	Test Engineer Productivity Value	M-36	Man-Months of Effort per Engineer per Month
TEPC	Productivity Constant for Test Engineers	M-36A	Man-Months of Effort per Engineer per Month

TABLE 6--Continued

Variable	Variable Name	Equation Describing	Dimension
PEMAP	Modified Aggregate Productivity of Production Engineers	M-37	Man-Months of Effort per Month
MMM	Man-Month Modifier	M-38	(None)
TMMM	Table of Man-Month Modifiers Versus Intrinsic Effort and Engineers in Production	M-39	(None)
PERAP	Raw Aggregate Productivity of Production Engineers	M-40	Man-Months of Effort per Month
PVEIT PVEXE PVMTE	Productivity Values for Production Engineers	M-41	Man-Months of Effort per Month per Engineer
PCEIT PCEXE PCMTE	Productivity Constants for Production Engineers	M-41A	Man-Months of Effort per Month per Engineer
INEFRQ	Intrinsic Effort Required	P-4	Man-Months of Effort
QUP	Quality Improvement Pressure	QM-9	(None)
PRP	Production Pressure	PM-13	(None)

In the development of the generalized software production process, in Chapter V, the resulting model showed that the determinants of both the software production rate and the rework discovery rate were the technology applied to these tasks and the productivity of the manpower assigned. At the top of Figure 7-9, the linkages between the manpower and productivity sector and the software production loop are shown. Both linkages are between the previously discussed rates and an aggregate productivity value for either production or test engineers.

The central portion of Figure 7-9 shows the allocation of engineers in each skill level to either testing or production, based upon the variable proportion of total engineers assigned to testing. Two factors affect the determination of this proportion: quality improvement pressure and production pressure. Quality improvement pressures within the organization are typically responded to by increasing the manpower allocated to related efforts, such as software testing. Production pressures within the organization are responded to similarly but by allocating increased manpower to direct production. The resulting interaction results in the determination of proportional assignment of engineers to both tasks, as described in Figure 7-10.

The productivity of individual test engineers is also affected by the counteracting influences of quality

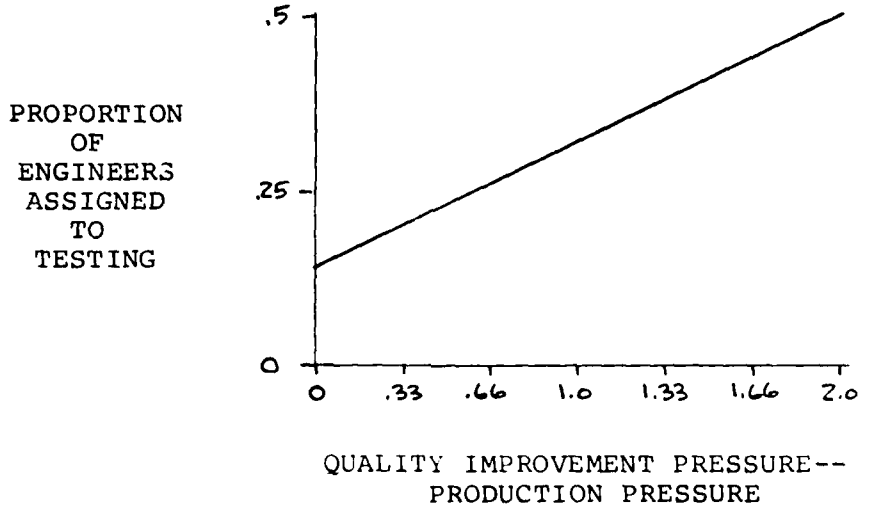


Fig. 7-10. Table of Proportion of Engineers in Testing versus Quality Improvement and Production Pressures

improvement and production pressures. Quality improvement pressure upon the engineer causes his concern to shift from the volume of software tested to attention to the completeness and correctness of his efforts, the result is typically a decrease in the volume of software tested. Production pressure, on the other hand, causes concern to shift oppositely towards the volume of software tested, and an increase in his productivity value. The maintenance of a balance is the concern of both production and quality management.

By summing the number of engineers at all skill levels assigned to testing, the total engineers in testing can be determined by subtracting the engineers at each skill level assigned to testing from the level of engineers

in the organization at each skill level, from Figure 7-8, a determination of the number of engineers at each skill level in production results. The total engineers in production results from the difference of total engineers in the organization and total engineers in testing.

The equations describing the productivity information network as presented thus far are listed below:

```
TEIT.K=(PEAT.K)(ENGIT.K)                    M-25,Auxillary
TEXE.K=(PEAT.K)(ENGEX.K)                    M-26,Auxillary
TMTE.K=(PEAT.K)(PMTE.K)                     M-27,Auxillary
TENGR.K=TEIT.K+TEXE.K+TMTE.K                M-30,Auxillary
PEIT.K=ENGIT.K-TEIT.K                       M-31,Auxillary
PEXE.K=ENGEX.K-TEXE.K                       M-32,Auxillary
PMTE.K=ENGMT.K-TMTE.K                       M-33,Auxillary
PENGR.K=ENGTOT.K-TENGR.K                    M-34,Auxillary
TEAP.K=(TENGR.K)(TEPV.K)                    M-35,Auxillary
TEPV.K=(TEPC)(QIP.K-1)                      M-36,Auxillary
TEPC=1.0                                    M-36A,Constant
```

The remainder of Figure 7-9 translates the numbers and skill levels of production engineers to the modified aggregate productivity affecting the software production rate. The productivity of each skill level of production engineer can be described as fluctuating over time around some base constant in response to system influences. The model establishes a productivity constant of 1 for experienced engineers, meaning that in one month such an engineer will produce one man-month of effort. The varying value of production pressure within the organization perturbs the productivity value of an experienced engineer, so that it varies within an established range. This range is bounded by some upper maximum representing

the maximal productivity performance of an engineer regardless of the magnitude of the pressure, and by some lower bound representing the minimal productivity performance of an engineer consistent with his self-image and professional motivation.

Because engineers in training and engineers assigned as managers and trainers contribute to a lesser degree to direct software production, their base productivity constants are established at a lower value: 0.4 and 0.7 man-months of effort, respectively. The varying value of production pressure, similarly perturbs the production values of such engineers within an established range. The composite of productivity value and number of engineers in each skill level produces a *raw aggregate productivity* for all production engineers.

The conceptual discussion in Chapter VI presented the problem of increasing communications difficulty among engineers working on a software project as the number of those engineers was increased. The influence of this difficulty was an increasing communications overhead and resulting decrease in direct software production. Figure 7-11 is used in the model to determine the value of this modifying influence as a function of the intrinsic size of the project and the number of production engineers involved. The product of this modifier and the raw

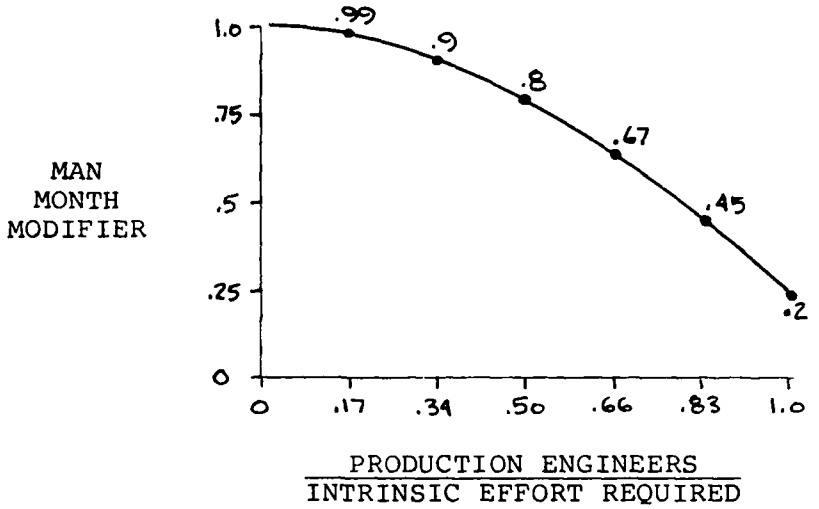


Fig. 7-11. Table of Man-Month Modifiers versus Total Production Engineers and Intrinsic Effort Required

aggregate productivity of production engineers results in the modified aggregate productivity affecting the software production rate.

The equations describing the remainder of the manpower and productivity sector are presented below:

```
PEMAP.K=(PERAP.K)(MMM.K)                          M-37,Auxillary
PERAP.K=(PEIT.K)(PVEIT.K)+(PEXE.K)                M-40,Auxillary
        (PVEXE.K)+(PMTE.K)(PVMTE.K)
PVEIT.K=(PCEIT)(PRP.K)                            M-41,Auxillary
PVEXE.K=(PCEXE)(PRP.K)                            M-41,Auxillary
PVMTE.K=(PCMTE)(PRP.K)                            M-41,Auxillary
PCEIT=0.4                                         M-41A,Constant
PCEXE=1.0                                         M-41A,Constant
PCMTE=0.7                                         M-41A,Constant
```

Quality Management Sector

The last sector model, the quality management sector, determines the flow of rework through the software production loop from undiscovered rework to discovered

rework to a return to work in progress. The error and discovery rates as controlled by an associated information network are the determinants of that flow. This model is presented in Figure 7-12.

Quality management continually monitors the level of discovered rework; comparing it to production management's estimate of work accomplished to arrive at a perceived error rate in software production. This perceived error rate is compared to an acceptable error rate established as system policy to determine the excess error rate. Pressure to improve the organization's quality effort results from a positive value for this excess error rate. This relationship is shown in Figure 7-13.

The improvement of the organization's software quality effort is the result of an increase in its level of software quality management capability, similar to the level of management capability to implement technology discussed in the technology sector model. This level of management capability is determined by the rate of improvement influenced by two factors: the quality improvement pressure and the proportion of funding requests for quality improvement approved.

The direction of effort in quality management is towards either quality in production or quality testing programs. Management policy determines the proportion of effort directed towards each, and that proportion

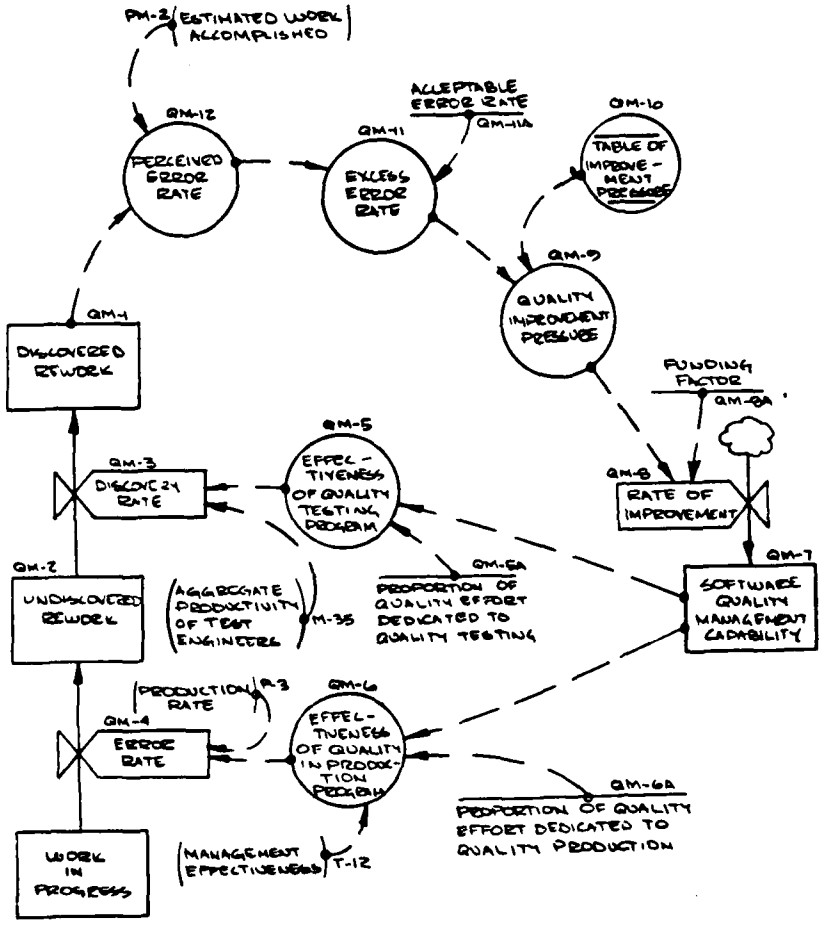


Fig. 7-12. Quality Management Sector

TABLE 7

QUALITY MANAGEMENT SECTOR VARIABLES

Variable	Variable Name	Equation Describing	Dimension
DISRE	Discovered Rework	QM-1	Man-Months of Effort
UDISRE	Undiscovered Rework	QM-2	Man-Months of Effort
RDISC	Rate of Discovery of Rework	QM-3	Man-Months of Effort per Month
RERR	Rate of Error in Software Production	QM-4	Man-Months of Effort per Month
TQEF	Effectiveness of Quality Testing Program	QM-5	(None)
FQTP	Proportion of Quality Effort Dedicated to Quality Testing	QM-5A	(None)
PQEF	Effectiveness of Quality in Production Program	QM-6	(None)
FQPP	Proportion of Quality Effort Dedicated to Quality in Production	QM-6A	(None)
SQMC	Software Quality Management Capability	QM-7	(None)
RQMIMP	Rate of Improvement in Software Quality Management Capability	QM-8	(None)
QMIFF	Quality Management Improvement Funding Factor	QM-8A	(None)
QIP	Quality Improvement Pressure	QM-9	(None)
TQIP	Table of Quality Improvement Pressure Versus Excess Error Rate	QM-10	(None)

TABLE 7--Continued

Variable	Variable Name	Equation Describing	Dimension
EER	Excess Error Rate	QM-11	Man-Months of Effort per Month
AER	Acceptable Error Rate	QM-11A	Man-Months of Effort per Month
PER	Perceived Error Rate	QM-12	Man-Months of Effort per Month
RPRO	Software Production Rate	P-3	Man-Months of Effort per Month
EWA	Estimated Work Accomplished	PM-2	Man-Months of Effort per Month
MGTEF	Management Effectiveness	T-12	(None)
PEAT	Aggregate Productivity of Test Engineers	M-35	Man-Months of Effort per Month

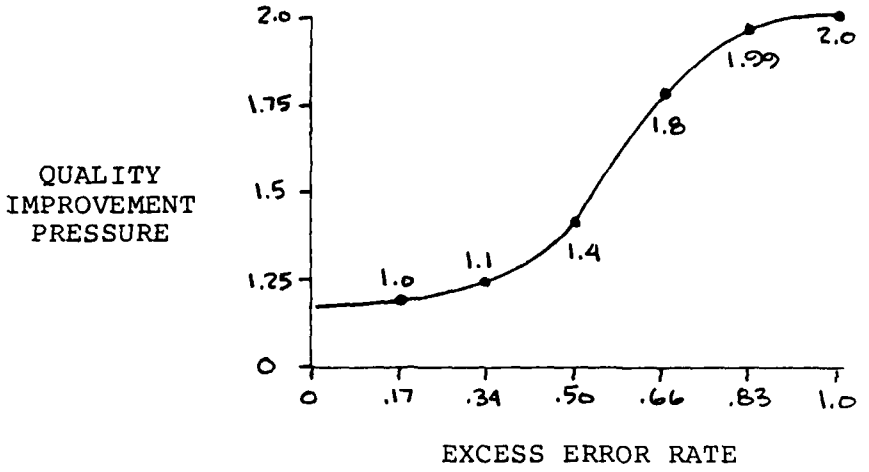


Fig. 7-13. Table of Quality Improvement Pressure versus Excess Error Rate

places a bound on the effectiveness of each program. The effectiveness of quality testing in conjunction with the aggregate productivity of test engineers determines the rework discovery rate. Similarly, the effectiveness of the quality in production emphasis in conjunction with the production rate determines the error rate.

The equations describing the quality management sector follow:

```
DISRE.K=DISRE.J+DT(RDISC.JK)                    QM-1,Level
UDISRE.K=UDISRE.J+DT(RERR.JK-RDISC.JK)          QM-2,Level
RDISC.KL=(TEAP.K)(TQEF.K)                       QM-3,Rate
RERR.KL=(1-PQEF.K)(RPRO.JK)                     QM-4,Rate
TQEF.K=(SQMC.K/TSOA.K)(FQTP)                    QM-5,Auxillary
FQTP=0.5                                        QM-5A,Constant
PQEF.K=(SQMC.K/TSOA.K)(FQPP)                    QM-6,Auxillary
FQPP=0.5                                        QM-6A,Constant
SQMC.K=SQMC.J+DT(RQIMP.JK)                      QM-7,Level
RQMIMP.KL=(QIP.K-1)(QMIFF)(SQMC.K)              QM-8,Rate
QMIFF=.75                                       QM-8A,Constant
EER.K=MAX((PER.K/AER)-1,0)                      QM-11,Auxillary
AER=.1                                          QM-11A,Constant
PER.K=FIFZE(0,(DISRE.K/EWA.K),EWA.K)            QM-12,Auxillary
```

Chapter Summation

This chapter concludes the third stage of the system dynamics study of weapon system software acquisition and support. As stated in previous chapters, the iterative nature of such work precludes a determination of definitive completion for the mathematical model at this point. The computerization of the model and the exploration of its behavior as a representation of the actual software acquisition and support process initiates a reexamination of the structures developed thus far as that behavior is analyzed.

The development of the conceptual model and its translation into a mathematical model represents the postulation of a hypothesis of the system structure producing the problematic system behavior. The computerization and testing of the mathematical model, which follows, corresponds to the testing of that postulated hypothesis. The results will serve to verify the utility of the model in not only increasing the general understanding of the complexities, interaction and dynamic behavior of the weapon system software acquisition and support system, but also as an analysis tool for the formulation and implementation of management policy.

VIII. Computerization, Verification and Experimentation

Introduction

As presented in Chapter III, the fourth stage of a system dynamics study of the weapon system software acquisition and support system is the computerization of the mathematical model and its use to generate simulated system behavior. In order to use the model for any meaningful analysis and improvement of that system some verification and validation process must be performed. Initial behavior exploration with the model should lead to acceptance or rejection of initial hypotheses about system structure and behavior. When an acceptable model structure has been developed, satisfactorily portraying system behavior, then adjustments to policy related parameters and structure of the model form the basis of experimentation. Such experimentation substantiates hypotheses developed from previous experiments about potential improvements in system behavior as a result of redesign of system policy and structure.

This chapter presents a discussion of the computerization of the weapon system software acquisition and support system model, an approach to verification and validation of that model, and a proposal for policy experimentation with the model.

Computerization

An assumption that characterizes mathematical and computer modelling of socio-technical systems is that in the aggregate human decisions and actions can be quantified into computer equations and that such equations can be grouped into representations or computer models of systems which are potentially better representations than any other as a basis for analysis and decision. Forrester postulated that individual and societal decisions must be made on the basis of some model, usually a *mental model*. This mental model he stated is the set of unexpressed assumptions and generalizations about the world which exists within each person's mind (Ref 18).

Complete knowledge upon which to base one's decisions and actions is not available. Only more or less simplified models based upon one's education, culture, and personal experience are available. Because decisions must be based upon some sort of incomplete and uncertain model, Meadows proposes that a computer model may be preferable to a mental model because:

1. It is precise and rigorous instead of ambiguous and unquantified,

2. It is explicit and can be examined by critics for inconsistency or error,

3. It can contain much more information than any single mental model,

4. It can proceed from assumptions to conclusions in a logical, error-free manner, and

5. It can easily be altered to represent different assumptions or alternate policies (Ref 28:27).

It is precisely these advantages of a computer model that support the verification, validation and experimentation with a computer model of the weapon system software acquisition and support system discussed later in this chapter. The initial development of such a computer model is very simply the translation of the mathematical model in Chapter VII into DYNAMO equations with the addition of a few language and machine specific control statements; followed by the machine-internal representation or computerization of those equations and statements. A complete and executable listing of this sequence is contained in Appendix A.

An Approach to Verification and Validation

The problems of verification and validation of computer simulation models constitute a major continuing debate in and out of the field of computer modelling. Shannon states that the task of verification is the insuring that the model behaves as the experimenter intends, while that of validation is the testing of agreement between the behavior of the model and that of the real system being represented (Ref 40:210). In both cases the establishment

of confidence in the model, both in conforming to the designer's intent and in replicating the system represented, becomes the major concern (Ref 29:310-320). Coyle summarizes both verification and validation as "the process by which we establish sufficient confidence in a model to be prepared to use it for some particular purpose [Ref 9:181]."

The verification and validation of the model developed in this study can be examined in light of its purpose. At the start of this investigation, the objective was stated as the presentation of a theory or model to support improved understanding of the complexities and interactions in the decision structure of weapon system software acquisition and support. This is in keeping with the nonexperimental mode of research, or with what Naylor describes as explanatory or positive analysis, where such a model must be subjected to direct questioning or empirical observation for verification or validation (Ref 29:315).

Recalling that the motivating thesis of this study was that the solution to the problems of weapon system software acquisition and support will result from the analysis and formulation of policy to guide and improve the performance of that management system, and that system dynamics is a policy-related tool, it becomes apparent that the only true test of this model's validity, theoretically possible, is to observe the actual system at a

suitable time after a policy change to make sure that performance (behavior) has indeed improved. Strictly speaking, as Coyle proposes, true validation is impossible and confidence in the model as meeting its purpose again becomes the objective of the validation and verification process. He proposes five questions to increase confidence and thus to indirectly assess model validity (Ref 9:182).

"Is the system boundary right?" is Coyle's first question. The model must include "those parts of the system, especially its controller, which can be changed to improve behavior [Ref 9:182]." The analysis of Chapter V is directed at providing a positive response to this first question. The contextual considerations are examined from a variety of views.

The second question concerns the reasonableness of the model and is simply stated as: "Are there any gross errors in the model?" A model which produces behavior which is conceptually impossible, or which is beyond all sense and reason for the system is obviously not valid (Ref 9:182). This question can only be answered after some initial behavioral exploration with the model. However, during system conceptualization and translation to a mathematical model, special attention to causal relationships and variable interactions minimizes the potential for such errors occurring. The detailed effort presented in

Chapters V, VI and VII was in part motivated by this question.

The appropriateness of the model's structure is addressed in the third question: "Is there correspondence between the model structure and that of the system?" Variables are checked for proper interconnections and decision functions and examined to ascertain that they reasonably reflect those actually used. This is very difficult to do. Rarely are data available to verify that the modelled decision function reflects the past. Even when such data are available, the best that can be done is to reject an obviously erroneous formulation (Ref 9:182-3).

In answering the third question it is necessary to examine the basis for selecting the primary components of the model: the production, production management, technology, manpower and productivity, and quality management sectors. The model deals with a common underlying structure of weapon system software acquisition and support, the production process for attaining and maintaining quality in the software product; and the influences upon the behavior of that structure. Again Chapters V and VI examine in depth those relationships. The correspondence between causal variables, their linkages and similar structures in the real system are explicitly developed.

A related question asks: "Are the model parameter values correct?" Do the values of model constants and

the dynamics of parameter tables accurately represent the corresponding, although unapparent values, in the real system? Coyle states that the general dynamics of the system are not usually affected very much by most of the parameters, provided they are within a fairly broad range. Only a few parameters will be more critical (Ref 9:183). The development of constants and tables within the weapon system software acquisition and support model, although somewhat intuitive, draws heavily upon the measurement of such relationships and the experience and judgement expressed in the literature.

The last question asked is: "Does the model reproduce system behavior?" Again the answer to this question is dependent upon initial behavioral exploration. The classical method of providing an answer is to compare time series data for the same variables from the model and the real system. The model fails validation if such values do not agree within statistical bounds. Coyle summarizes a variety of difficulties with this technique, all generally related to the incompleteness of historical data. The key test, and that to be applied in future research to the model developed in this study, is the consensus of actors within the real system that the concepts structuring the model are indeed apparent in the real system (Ref 9:183-4).

The verification and validation of the weapon system software acquisition and support system model

developed in this study was not a consideration left to the last phases of the work, it was instead considered and built into the model by keeping in mind the previous five questions throughout model development.

A View Towards Experimentation

The ultimate goal of a system dynamics study of the weapon system software acquisition and support system is the implementation of policy recommendations formulated to guide and improve the performance of that system. Such policy recommendations are the result of a shift from a nonexperimental to an experimental mode of research where the model serves as a means to an end rather than the end product of the research.

Once some agreement or acceptance of the causes of a problem or the nature of the system generating it, as engendered in the system model, is reached, policy formulation and implementation can begin. Broad policy choices can be evaluated and compared to identify possible tradeoffs or synergies (Ref 28:28-29). These policy choices are considered through the design and execution of model experiments; the results of which, when properly analyzed, serve to predict the potentially successful of a number of policies and policy combinations.

In the weapon system software acquisition and support process, such policies revolve around a variety

of factors, including the acquisiton and allocation of manpower and technology resources; funding of improvements to technology, technology implementation, and quality management; and the proportion of effort directed between apparently competing activities such as testing and production, among others. The success of policies related to these factors or the response of the model can be represented in terms of weapon system effectiveness as discussed in Chapter V.

This testing of potential policies, described above essentially represents the transition from policy formulation to implementation. However, equally important is the transition from policy analysis as initiated by the efforts of this research to policy formulation. This is characterized by model testing to develop confidence and refutability and to identify especially sensitive parameters as candidates for increased concern in model refinement and as *probe* points for the formulation of policy.

The efforts of policy experimentation are constituted within the later phases of a system dynamics study. Those parts of such a study are essentially unreached as of yet by this research. The next chapter, and the last in this report, examines this area as a recommendation for future research in addition to other proposals to advance the operationalization of policy recommendations to grow from this research.

IX. Conclusions and Recommendations

Let no man say that I have said nothing new; the arrangement of the material is new. Just as the words differently arranged form different thoughts.

— Blaise Pascal

The goal of this study was the proposition of a new way of thinking about, studying and managing weapon system software acquisition and support: an approach based upon the systems nature of the process, acknowledging the process flow and feedback controlling aspects of such projects. The research presented thus far is conceptual in nature and not heavily reliant upon the quantification of variables and parameters. Even without such fully developed quantitative information it can be a useful tool for understanding and policy development.

It is apparent from the magnitude of cost and performance failures presented in Chapter I that decisions are being made without adequate information about fundamental system relationships. Managers aware of system decisional structures will most assuredly be better able to develop adequate information to support policy development and decision making. In the end, the goal should be, as Forrester states, *enterprise design* of an informational

and decisional structure to support the implementation of experimentally determined policy alternatives.

The shortcomings of this research have been implicitly presented in previous chapters, but will be reiterated here, as these weaknesses form the basis for proposed future research. The incompleteness in accomlishing the full cycle of system dynamics inquiry (see Figure 2-1) is apparent in this study. However, the fact that such a method or analysis view has not before been applied to the system investigated helps to bring acceptance to that work which is presented. An appropriate point of resolution was reached with the conclusion of the nonexperimental mode of research by this study. The full importance of this research will only be made apparent with the completion of the experimental mode and the implementation of policy improvements in the real system.

As touched upon in Chapter VIII, a major method of confidence building (validation) in system dynamics modelling is the consensus of real system managers with the concepts developed in constituting the conceptual model or structural theory of the system. Expert opinion was solicited early in this research process, and forms a major portion of the intuitive, experiential and judgemental influences upon the theory of structure presented. Before proper policy experimentation can be initiated the consensus of policy makers within the real system must be

obtained to verify that the individual linkages and collective decision structures in the model do indeed represent those of the real system. Their disagreement with the model will form the basis for continued refinement. The advantages of such refinement comes in implementation of experimentally formulated policy alternatives, because the results of such experiments are founded in the agreement of the policy makers with the model's description.

As implied above the lack of a design for the execution of policy experimentation, does not allow full comprehension of the potential of such a policy tool, as the model presented. Such an approach should form the major thrust of any future research.

Lastly, as in almost all system dynamics studies, the impact of supporting data upon study results is important to consider. The classical approach in modelling is the application of sophisticated statistical procedures to historical data concerning the time series behavior of the system and model variables to produce parameter and constant values for inclusion in the model. According to Coyle, this is rarely possible, as the data are not usually available. Even when data can be found, they relate only to system states and rarely are the policies by which these states were controlled known (Ref 9:183). An initial result of this study is the identification of that data which needs to be collected and summarized in the future

to support improvement of the model's performance as a policy analysis and formulation tool.

The link between data modelling and experimentation is sensitivity analysis. Most information-feedback systems are insensitive to minor changes within a wide range for the majority of parameter values (Ref 9:183). However, the identification of sensitive parameters in affecting system behavior is of utmost importance and should be an initial point of experimentation in future research.

Besides preliminary use in general understanding of the complexities and interactions within the weapon system software acquisition and support system, the ultimate use for a model of that system will be as a core or model base for a decision support system. In such a system the intuition, experience and judgement of policy and decision makers at all levels of the acquisition and support system are amplified and aided by, among other things, a computer model exhibiting the advantages discussed in Chapter VIII. The overall synergistic effect is the selection of policy and the making of decisions far improved from those supported by only the mental model of the human participant, or those supported by the mechanistic calculations of a computing system not yet *evolved* to employ the full range of perceptual abilities of the man. Recalling the previous discussions of Ashby's Law of Requisite

Variety the computer model of the acquisition and support process maintained within a decision support system serves a similar function as an embedded computer system in an aircraft, in absorbing some of the variety of the situation presented to the policy maker.

Policy as the criteria for making the multitude of operating decisions within a system as complex as that investigated by this research, is itself the result of a dynamic information feedback process. A conceptual view of this process and the role of a computer model such as presented here is illustrated in Figure 9-1. The computer model forms the model base in the figure. It is viewed as continually evolving as the analysis of policy problems places more demands upon it, as experimentation provides insight and refinement, and as data collection and modelling improves the determination of model parameters.

The intuition, experience and judgement of the policy maker control the activities noted on the flows from stage to stage in the process. The process is initiated as the policy maker's perception of the magnitude of a policy problem crosses a threshold triggering his direction of an analysis. The data and model bases are employed to determine the systemic causes of the problematic behavior. The formulation and evaluation of policy adjustments to reestablish the performance of the system is initiated when an acceptable model representation of the present and

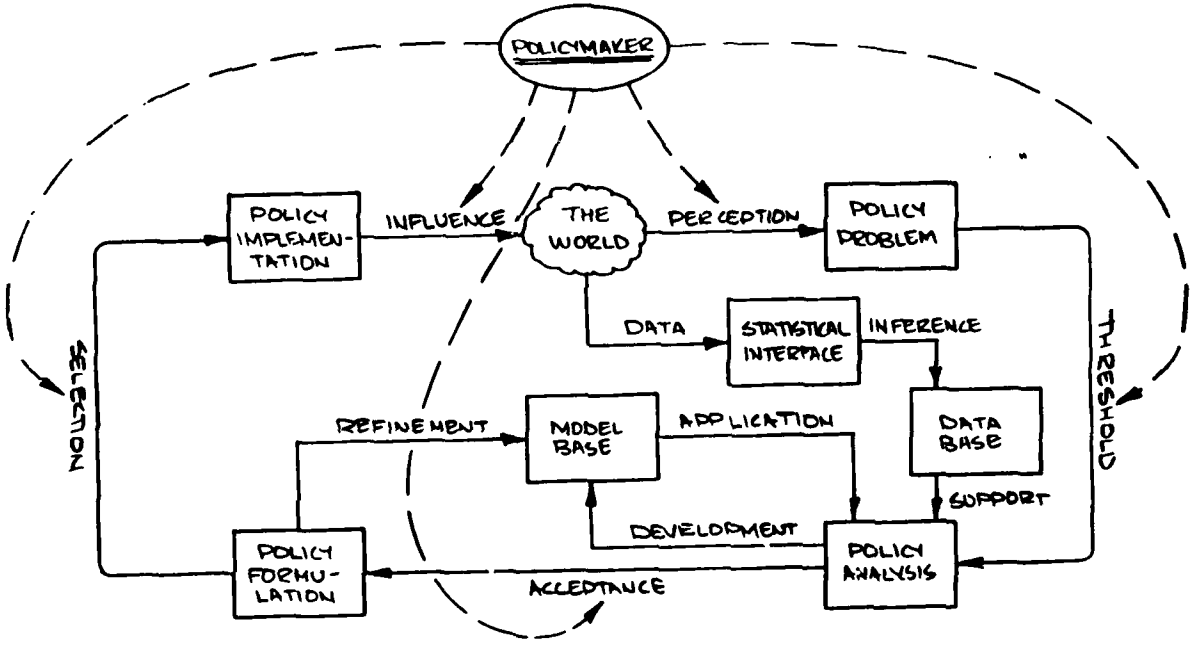


Fig. 9-1. A Conceptual View of Policy Oriented Decision Support

potential behavior and states of the system is obtained. This evaluation or experimentation is concluded with the selection by the policymaker of the policy adjustment to be implemented. The implementation of policy influences the ever-changing state of the world, producing new data for collection and filtering. The new state of the world and its associated behavior will, hopefully, be an improvement because of the predictive capability of the model, but may instead result in a policy maker's perception of new problems of differing magnitudes and in different areas to repeat the cycle.

In closing, the research presented, when examined in the context of Figure 9-1, can be seen as a small beginning in correcting the difficulties of cost and performance in acquiring and supporting weapon system software. However small that beginning may be, it is the first step in constructing a new approach to such problems. The importance of continuing such research cannot be overemphasized. The modelling and world views developed in such research are quickly transferable to the solution of many more, and certainly more critical, problems of human endeavor.

Bibliography

1. Acker, D. "The Maturing of the DOD Acquisition Process," Defense Systems Management Review, 3(3): 7-77 (Summer 1980).

2. Alberts, David S. The Economics of Software Quality Assurance. MTR-5272; Bedford, Mass.: The Mitre Corporation, December 1975.

3. Anderson, David F. and George P. Richardson. "Toward a Pedagogy of System Dynamics" in TIMS Studies in the Management Sciences, Volume 14, System Dynamics, edited by Augusto A. Legasto, Jr., et al. New York: North-Holland Publishing Company, 1980.

4. Beer, Stafford. Decision and Control. New York: John Wiley and Sons, Inc., 1966.

5. Boehm, Barry W. "Software and its Impact: A Quantitative Assessment," Datamation (May 1973).

6. Brooks, Frderick D., Jr. The Mythical Man-Month: Essays in Software Engineering. Reading, Mass.: Addison-Wesley Publishing Company, 1975.

7. Cooper, Kenneth G. et al. "Naval Ship Production: A Claim Settled and a Framework Built," Interfaces, The Institute of Management Sciences, 10(6): 20-36 (December 1980).

8. Corder, David R. "Oklahoma City Air Logistics Center Approach to Embedded Computer Systems Support," Proceedings of the National Aerospace Electronics Conference. 143-7. Dayton, Ohio: Dayton Section, IEEE, May 1980.

9. Coyle, R. G. Management System Dynamics. New York: John Wiley and Sons, Inc., 1977.

10. Department of the Air Force. AFR 800-14, Volume I, "Management of Computer Resources in Systems." Washington, D.C.: September 1975.

11. _____. AFR 800-14, Volume II, "Acquisition and Support Procedures for Computer Resources in Systems." Washington, D.C.: September 1975.

12. Department of Defense. Directive 5000.1, "Major System Acquisition." Washington, D.C.: January 1977.

13. _____. Directive 5000.29, "Management of Computer Resources in Major Defense Systems." Washington, D.C.: April 1976.

14. _____. Instruction 5000.1, "Acquisition of Major Defense Systems." Washington, D.C.

15. _____. Office of the Undersecretary of Defense for Research and Engineering. "Embedded Computer Resources and the DSARC Process: A Guidebook." Washington, D.C.: 1981.

16. DeRoze, Barry C. "An Introspective Analysis of DOD Weapon System Software Management," Defense Management Journal, October 1975.

17. Fisher, Matthew J. and William R. Light, Jr. "Definitions in Software Quality Management" in Software Quality Management, edited by Fisher and Cooper. New York: Petrocelli Books, Inc., 1979.

18. Forrester, Jay W. Industrial Dynamics. Cambridge, Mass: The M.I.T. Press, 1961.

19. _____. "Industrial Dynamics--After the First Decade," Management Science, 14(7): 398-415 (March 1968).

20. _____. "System Dynamics--Future Opportunities" in TIMS Studies in the Management Sciences, edited by Augusto A. Legasto, Jr. et al. New York: North-Holland Publishing Company, 1980.

21. Green, Robert P., Jr. Deputy Chief. Personal interview. Materials Maintenance Engineering, Warner-Robins Air Logistics Center, Warner Robins AFB, Georgia, September 29, 1981.

22. Karns, Brigadier General Robert C. "The Significance of Software Costs," Proceedings of the National Aerospace Electronics Conference. 662. Dayton, Ohio: Dayton Section, IEEE, May 1981.

23. Katz, Abraham. An Operations Analysis of an Electronic Systems Firm. Unpublished SM Thesis. Massachusetts Institute of Technology, Cambridge, Mass., 1958.

24. Knight, B. M. "Software Quality and Productivity," Defense Systems Management Review, 1(7-8): 54-65 (Autumn 1978).

25. Legasto, Augusto A., Jr., Jay W. Forrester and James M. Lyneis, editors. TIMS Studies in the Management Sciences, Volume 14, System Dynamics. New York: North-Holland Publishing Company, 1980.

26. Marsh, Lieutenant General Robert T. "The Impact of Today's Electronics Technology on Systems Acquisition" quoted in "Software Support Equation of Accountability--A Challenge" by John D. R. Ferrell, Proceedings of the National Aerospace Electronics Conference. 152-6. Dayton, Ohio: Dayton Section, IEEE, May 1980.

27. McCall, James A. "An Introduction to Software Quality Metrics" in Software Quality Management, edited by Fisher and Cooper. New York: Petrocelli Books, Inc., 1979.

28. Meadows, Donella H. "The Unavoidable A Priori" in Elements of the System Dynamics Method edited by Jorgen Randers. Cambridge, Mass.: The M.I.T. Press, 1980.

29. Naylor, Thomas H. et al. Computer Simulation Techniques. New York: John Wiley and Sons, Inc., 1966.

30. Pugh, Alexander L. III. DYNAMO User's Manual. Cambridge, Mass.: The M.I.T. Press, 1976.

31. Roberts, Edward B. "The Design of Management Control Systems" in Managerial Applications of System Dynamics, edited by Roberts. Cambridge, Mass.: The M.I.T. Press, 1978.

32. _____. The Dynamics of Research and Development. New York: Harper and Row, Inc., 1964.

33. _____. "System Dynamics--An Introduction" in Managerial Applications of System Dynamics edited by Roberts. Cambridge, Mass.: The M.I.T. Press, 1978.

34. Shannon, Robert E. Systems Simulation: The Art and the Science. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

35. Software Acquisition Guidebook: Life Cycle Events. ESD-TR-77-22; Hanscom AFB, Mass.: Electronic Systems Division, Air Force Systems Command, 1977.

36. Software Acquisition Guidebook: Regulations, Specifications and Standards. ASD-TR-78-6; Wright-Patterson AFB, Ohio: Aeronautical Systems Division, Air Force Systems Command, 1977.

37. "Software Improvement Plan Pressed," Aviation Week and Space Technology, 104(14): 41-3 (April 5, 1976).

38. Starr, Patrick J. "Modelling Issues and Decisions" in TIMS Studies in the Management Sciences, Volume 14, System Dynamics edited by Augusto A. Legasto, Jr. et al. New York: North-Holland Publishing Company, 1980.

39. Trainor, W. L. "Software for Avionics: An Overview," Proceedings of the National Aerospace Electronics Conference. 227-233. Dayton, Ohio: Dayton Section, IEEE, June 1975.

40. _____. "Software from Satan to Saviour," Proceedings of the National Aerospace Electronics Conference. 22-9. Dayton, Ohio: Dayton Section, IEEE, May 1973.

Appendix A

DYNAMO Model Equations

```
*          WEAPON SYSTEM SOFTWARE ACQUISITION MODEL                    *
*  ********** ********** ********** ********** ********** ********** **
*  ********** ********** ********** ********** ********** ********** **


*  PRODUCTION SECTOR
*  ****************

L  WIP.K=WIP.J+(DT)(RDISC.JK-PROADJ.JK-RPRO.JK)                          P-1
N  WIP=INEFRO                                                           P-1N
L  WAC.K=WAC.J+(DT)(RPRO.JK-RERR.JK)                                    P-2
N  WAC=0                                                                P-2N
R  RPRO.KL=(UTE.K)(PEMAP.K)                                             P-3
N  INEFRO=INSZPJ/TSOA                                                   P-4N
*  INSZPJ=NORMN(50000,10000)                                            P-5N
N  INSZPJ=100000
      WIP - WORK IN PROGRESS (MAN-MONTHS OF EFFORT)
      INEFRO - INTRINSIC EFFORT REQUIRED (MAN-MONTHS OF EFFORT)
      WAC - WORK ACCOMPLISHED (MAN-MONTHS OF EFFORT)
      RDISC - RATE OF DISCOVERY OF REWORK (MAN-MONTHS OF
              EFFORT PER MONTH)
      PROADJ - RATE OF ADJUSTMENTS TO REQUIREMENTS (MAN-MONTHS
               OF EFFORT PER MONTH)
      RPRO - RATE OF PRODUCTION (MAN-MONTHS OF EFFORT PER MONTH)
      RERR - RATE OF ERRORS IN PRODUCTION (MAN-MONTHS OF EFFORT
             PER MONTH)
      UTE - UTILIZED TECHNICAL EFFECTIVENESS
      PEMAP - MODIFIED AGGREGATE PRODUCTIVITY OF PRODUCTION
              ENGINEERS (MAN-MONTHS OF EFFORT PER MONTH)
      INSZPJ - INTRINSIC SIZE OF THE PROJECT
      TSOA - STATE OF THE ART IN SOFTWARE DEVELOPMENT TECHNOLOGY


*  MANPOWER AND PRODUCTIVITY SECTOR
*  ********** ********** ********** **

L  ENGPA.K=ENGPA.J+(DT)(REJW.JK+RELOA.JK+REITL.JK+REXEL.JK
X  +RMTEL.JK-RELW.JK-REJOA.JK-RHIRE.JK)                                 M-1
N  ENGPA=50                                                             M-1N
L  ENGOA.K=ENGOA.J+(DT)(REJOA.JK-RELOA.JK)                              M-2
N  ENGOA=100                                                            M-2N
R  REJW.KL=(FNEP)(ENGPA.K)                                              M-3
C  FNEP=.019                                                            M-3A
R  RELW.KL=(FWFA)(ENGPA.K)                                              M-4
C  FWFA=.019                                                            M-4A
R  RELOA.KL=(FOAA)(ENGOA.K)                                             M-5
C  FOAA=.019                                                            M-5A
R  REJOA.KL=(FOAR)(ENGOA.K)                                             M-6
C  FOAR=.021                                                            M-6A
R  RHIRE.KL=MIN(DFHF.K,MEHR.K)                                          M-7
R  REITL.KL=(FEITA)(ENGIT.K)                                            M-8
C  FEITA=.01                                                            M-8A
R  REXEL.KL=(FEXEA)(ENGEX.K)                                            M-9
```

```
C  FEXEA=.025                                                    M-9A
R  RMTEL.KL=(FMTEA)(ENGMT.K)                                     M-10
C  FMTEA=.01                                                     M-10A
      ENGPA - POOL OF AVAILABLE ENGINEERS (ENGINEERS)
      ENGOA - ENGINEERS IN OTHER ACTIVITIES (ENGINEERS)
      REJW - RATE ENGINEERS JOIN THE WORKFORCE
             (ENGINEERS PER MONTH)
      RELW - RATE ENGINEERS LEAVE THE WORKFORCE
             (ENGINEERS PER MONTH)
      REJOA - RATE AVAILABLE ENGINEERS JOIN OTHER ACTIVITIES
              (ENGINEERS PER MONTH)
      RELOA - RATE ENGINEERS LEAVE OTHER ACTIVITIES
              (ENGINEERS PER MONTH)
      REITL - RATE ENGINEERS IN TRAINING LEAVE THE ORGANIZATION
              (ENGINEERS PER MONTH)
      REXEL - RATE EXPERIENCED ENGINEERS LEAVE THE ORGANIZATION
              (ENGINEERS PER MONTH)
      RMTEL - RATE ENGINEER MANAGERS/TRAINERS LEAVE THE
              ORGANIZATION (ENGINEERS PER MONTH)
      RHIRE - RATE OF ENGINEER HIRING (ENGINEERS PER MONTH)
      FNEP - NEW ENGINEER PRODUCTION FACTOR
      FWFA - WORKFORCE ATTRITION FACTOR
      FOAR - RECRUITING BY OTHER ACTIVITIES FACTOR
      FOAA - ATTRITION IN OTHER ACTIVITIES FACTOR
      FEITA - ENGINEER IN TRAINING ATTRITION FACTOR
      FEXEA - EXPERIENCED ENGINEER ATTRITION FACTOR
      FMTEA - ENGINEER MANAGER/TRAINER ATTRITION FACTOR
      DEHR - DESIRED ENGINEER HIRING RATE (ENGINEERS PER MONTH)
      MEHR - MAXIMUM ENGINEER HIRING RATE (ENGINEERS PER MONTH)
      ENGIT - ENGINEERS IN TRAINING (ENGINEERS)
      ENGEX - EXPERIENCED ENGINEERS (ENGINEERS)
      ENGMT - ENGINEERS ASSIGNED AS MANAGERS/TRAINERS (ENGINEERS)
L  ENGBR.K=ENGBR.J+(DT)(RHIRE.JK-ENGJO.JK)                       M-11
N  ENGBR=0                                                       M-11N
R  ENGJO.KL=DELAY3(RHIRE.JK,DRCTG)                               M-12
C  DRCTG=3                                                       M-12A
      ENGBR - ENGINEERS BEING RECRUITED (ENGINEERS)
      RHIRE - RATE OF ENGINEER HIRING (ENGINEERS PER MONTH)
      ENGJO - RATE RECRUITED ENGINEERS JOIN THE ORGANIZATION
              (ENGINEERS PER MONTH)
      DRCTG - DELAY IN RECRUITING ENGINEERS (MONTHS)
L  ENGIT.K=ENGIT.J+(DT)(ENGJO.JK-ENGCT.JK-REITL.JK)              M-13
N  ENGIT=0                                                       M-13N
R  ENGCT.KL=DELAY3(ENGJO.JK,DTRNG)                               M-14
C  DTRNG=6                                                       M-14A
      ENGIT - ENGINEERS IN TRAINING (ENGINEERS)
      ENGJO - RATE ENGINEERS JOIN THE ORGANIZATION
              (ENGINEERS PER MONTH)
      ENGCT - RATE ENGINEERS COMPLETE TRAINING
              (ENGINEERS PER MONTH)
      REITL - RATE ENGINEERS IN TRAINING LEAVE THE ORGANIZATION
              (ENGINEERS PER MONTH)
      DTRNG - DELAY IN TRAINING ENGINEERS (MONTHS)
```

```
A  DEHR.K=ENGAP.K/DTREQ                                              M-15
C  DTREQ=1.5                                                        M-15A
A  ENGAP.K=MAX((ENGMA-ENGEN.K),0)                                    M-16
C  ENGMA=15                                                         M-16A
A  ENGEN.K=ENGTOT.K+(DT)(RHIRE.JK)                                   M-17
N  ENGEN=LEI                                                         M-17N
C  LEI=10                                                           M-17A
      DEHR - DESIRED ENGINEER HIRING RATE (ENGINEERS PER MONTH)
      ENGAP - ENGINEER GAP AT THE ORGANIZATION (ENGINEERS)
      DTREQ - DELAY IN TRANSMITTING REQUIREMENTS FOR ENGINEERS
              (MONTHS)
      ENGMA - ENGINEERING MANPOWER AUTHORIZED (ENGINEERS)
      ENGEN - EXPECTED NUMBER OF ENGINEERS (ENGINEERS)
      ENGTOT - TOTAL ENGINEERS IN THE ORGANIZATION (ENGINEERS)
      RHIRE - RATE OF ENGINEER HIRING (ENGINEERS PER MONTH)
      LEI - LEVEL OF ENGINEERS INITIALLY (ENGINEERS)
L  ENGTOT.K=ENGTOT.J+(DT)(ENGJO.JK-REITL.JK
X  -REXEL.JK-RMTEL.JK)                                               M-18
N  ENGTOT=LEI                                                        M-18N
A  MEHR.K=MAX((MNEIT.K-ENGIT.K)/DTREQ,0)                             M-19
A  MNEIT.K=(TSPD)(ENGMT.K+ENGEX.K)                                   M-20
C  TSPD=.3                                                          M-20A
L  ENGEX.K=ENGEX.J+(DT)(ENGCT.JK-REXEL.JK-RREA.JK)                   M-21
N  ENGEX=LEI                                                         M-21N
L  ENGMT.K=ENGMT.J+(DT)(RREA.JK-RMTEL.JK)                            M-22
N  ENGMT=0                                                           M-22A
R  RREA.KL=(DEAMT.K-ENGMT.K)/DREA                                    M-23
C  DREA=1                                                           M-23A
A  DEAMT.K=(FDRMT)(ENGTOT.K)                                         M-24
C  FDRMT=.2                                                         M-24A
      ENGTOT - TOTAL ENGINEERS IN THE ORGANIZATION (ENGINEERS)
      ENGJO - RATE ENGINEERS JOIN THE ORGANIZATION (ENGINEERS
              PER MONTH)
      REITL - RATE ENGINEERS IN TRAINING LEAVE THE ORGANIZATION
              (ENGINEERS PER MONTH)
      REXEL - RATE EXPERIENCED ENGINEERS LEAVE THE ORGANIZATION
              (ENGINEERS PER MONTH)
      RMTEL - RATE ENGINEER MANAGERS/TRAINERS LEAVE THE
              ORGANIZATION (ENGINEERS PER MONTH)
      MEHR - MAXIMUM ENGINEER HIRING RATE (ENGINEERS PER MONTH)
      DTREQ - DELAY IN TRANSMITTING REQUIREMENTS FOR ENGINEERS
              (MONTHS)
      MNEIT - MAXIMUM NUMBER OF ENGINEERS IN TRAINING (ENGINEERS)
      ENGIT - ENGINEERS IN TRAINING (ENGINEERS)
      ENGEX - EXPERIENCED ENGINEERS (ENGINEERS)
      ENGMT - ENGINEERS ASSIGNED AS MANAGERS/TRAINERS (ENGINEERS)
      TSPD - TRAINEES PER STAFF DESIRED (ENGINEERS)
      ENGCT - ENGINEERS COMPLETING TRAINING (ENGINEERS)
      RREA - RATE OF REASSIGNMENT OF ENGINEERS AS
             MANAGERS/TRAINERS (ENGINEERS PER MONTH)
      LEI - LEVEL OF ENGINEERS INITIALLY (ENGINEERS)
      RMTEL - RATE ENGINEER MANAGERS/TRAINERS LEAVE THE
              ORGANIZATION (ENGINEERS PER MONTH)
```

```
        DEAMT - DESIRED ENGINEERS ASSIGNED AS MANAGERS/TRAINERS
                (ENGINEERS)
        DREA - DELAY IN REASSIGNING ENGINEERS AS MANAGERS/TRAINERS
               (MONTHS)
        FDEMT - DESIRED RATIO OF MANAGERS/TRAINERS TO TOTAL
                ENGINEERS
A   TEIT.K=(PEAT.K)(ENGIT.K)                                          M-25
A   TEXE.K=(PEAT.K)(ENGEX.K)                                          M-26
A   TMTE.K=(PEAT.K)(ENGMT.K)                                          M-27
A   PEAT.K=TABHL(TPEAT,(QIP.K-PRP.K),0,2.04,.34)                     M-28
T   TPEAT=.15/.21/.26/.32/.37/.43/.5                                  M-29
A   TENGR.K=TEIT.K+TEXE.K+TMTE.K                                     M-30
        TEIT - TEST ENGINEERS IN TRAINING  (ENGINEERS)
        TEXE - EXPERIENCED ENGINEERS IN TESTING (ENGINEERS)
        TMTE - ENGINEER MANAGERS/TRAINERS IN TESTING (ENGINEERS)
        PEAT - PROPORTION OF ENGINEERS ASSIGNED TO TESTING
        ENGIT - ENGINEERS IN TRAINING (ENGINEERS)
        ENGEX - EXPERIENCED ENGINEERS (ENGINEERS)
        ENGMT - ENGINEERS ASSIGNED AS MANAGERS/TRAINERS
                (ENGINEERS)
        TPEAT - TABLE OF PROPORTION OF ENGINEERS ASSIGNED TO
                TESTING VERSUS QUALITY AND PRODUCTION PRESSURES
        TENGR - ENGINEERS IN TESTING (ENGINEERS)
A   PEIT.K=ENGIT.K-TEIT.K                                            M-31
A   PEXE.K=ENGEX.K-TEXE.K                                            M-32
A   PMTE.K=ENGMT.K-TMTE.K                                            M-33
A   PENGR.K=ENGTOT.K-TENGR.K                                         M-34
        PEIT - PRODUCTION ENGINEERS IN TRAINING (ENGINEERS)
        PEXE - EXPERIENCED ENGINEERS IN PRODUCTION (ENGINEERS)
        PMTE - ENGINEER MANAGERS/TRAINERS IN PRODUCTION (ENGINEERS)
        PENGR - ENGINEERS IN PRODUCTION (ENGINEERS)
        ENGIT - ENGINEERS IN TRAINING (ENGINEERS)
        ENGEX - EXPERIENCED ENGINEERS (ENGINEERS)
        ENGMT - ENGINEERS ASSIGNED AS MANAGERS/TRAINERS (ENGINEERS)
        TEIT - TEST ENGINEERS IN TRAINING (ENGINEERS)
        TEXE - EXPERIENCED ENGINEERS IN TESTING (ENGINEERS)
        TMTE - ENGINEER MANAGERS/TRAINERS IN TEST (ENGINEERS)
        TENGR - ENGINEERS IN TESTING (ENGINEERS)
A   TEAP.K=(TENGR.K)(TEPV.K)                                          M-35
A   TEPV.K=(TEPC)(QIP.K-1)                                           M-36
C   TEPC=1.                                                          M-36A
        TEAP - AGGREGATE PRODUCTIVITY OF TEST ENGINEERS
               (MAN-MONTHS OF EFFORT PER MONTH)
        TENGR - ENGINEERS IN TESTING (ENGINEERS)
        TEPV - TEST ENGINEER PRODUCTIVITY VALUE
               (MAN-MONTHS OF EFFORT PER ENGINEER PER MONTH)
        TEPC - PRODUCTIVITY CONSTANT FOR TEST ENGINEERS
               (MAN-MONTHS OF EFFORT PER ENGINEER PER MONTH)
        QIP - QUALITY IMPROVEMENT PRESSURE
        PRP - PRODUCTION PRESSURE
A   PEMAP.K=(PEAP.K)(MMM.K)                                          M-37
A   MMM.K=TABHL(TMMM,(PENGR.K/INEFRO),0,1.12,.17)                   M-38
T   TMMM=1/.99/.9/.8/.67/.45/.2                                      M-39
```

```
A   PERAP.K=(PEIT.K)(PVEIT.K)+(PEXE.K)(PVEXE.K)
X   +(PMTE.K)(PVMTE.K)                                              M-40
A   PVEIT.K=(PCEIT)(PRP.K)                                          M-41
A   PVEXE.K=(PCEXE)(PRP.K)                                          M-41
A   PVMTE.K=(PCMTE)(PRP.K)                                          M-41
C   PCEIT=0.4                                                       M-41A
C   PCEXE=1.0                                                       M-41A
C   PCMTE=0.7                                                       M-41A
      PEMAP - MODIFIED AGGREGATE PRODUCTIVITY OF PRODUCTION
              ENGINEERS (MAN-MONTHS OF EFFORT PER MONTH)
      PERAP - RAW AGGREGATE PRODUCTIVITY OF PRODUCTION ENGINEERS
              (MAN-MONTHS OF EFFORT PER MONTH)
      MMM - MAN-MONTH MODIFIER
      TMMM - TABLE OF MAN-MONTH MODIFIERS VERSUS INTRINSIC EFFORT
             REQUIRED AND ENGINEERS IN PRODUCTION
      PENGR - ENGINEERS IN PRODUCTION (ENGINEERS)
      INEFRQ - INTRINSIC EFFORT REQUIRED (MAN-MONTHS OF EFFORT)
      PEIT - PRODUCTION ENGINEERS IN TRAINING (ENGINEERS)
      PEXE - EXPERIENCED ENGINEERS IN PRODUCTION (ENGINEERS)
      PMTE - ENGINEER MANAGERS/TRAINERS IN PRODUCTION
             (ENGINEERS)
      PVEIT - PRODUCTIVITY VALUE FOR PRODUCTION ENGINEERS IN
              TRAINING (MAN-MONTHS OF EFFORT PER ENGINEER PER MONTH)
      PVEXE - PRODUCTIVITY VALUE FOR EXPERIENCED ENGINEERS IN
              PRODUCTION (MAN-MONTHS OF EFFORT PER ENGINEER PER
              MONTH)
      PVMTE - PRODUCTIVITY VALUE FOR ENGINEER MANAGERS/TRAINERS
              IN PRODUCTION (MAN-MONTHS OF EFFORT PER ENGINEER
              PER MONTH)
      PRP - PRODUCTION PRESSURE


*   TECHNOLOGY SECTOR
*   *****************

L   TSOA.K=TSOA.J+(DT)(RTIMP.JK)                                    T-1
N   TSOA=100                                                        T-1N
R   RTIMP.KL=(TPRES.K)(TIFF)(TSOA.K)                               T-2
C   TIFF=.75                                                        T-2A
A   DSOA.K=DLINF3(TSOA.K,DTECH)                                     T-3
C   DTECH=6                                                         T-3A
A   ATE.K=DSOA.K/TSOA.K                                             T-4
      TSOA - STATE OF THE ART IN SOFTWARE DEVELOPMENT TECHNOLOGY
      RTIMP - RATE OF IMPROVEMENT IN TECHNOLOGY STATE OF THE ART
      TPRES - PRESSURE FOR IMPROVEMENT IN TECHNOLOGY
      TIFF - TECHNOLOGY IMPROVEMENT FUNDING FACTOR
      DSOA - DELAYED STATE OF THE ART TECHNOLOGY
      DTECH - DELAY IN ACQUIRING STATE OF THE ART TECHNOLOGY
              (MONTHS)
      ATE - ACTUAL TECHNICAL EFFECTIVENESS
A   TGAP.K=MAX((IEFRQ.K-ATE.K),0)                                   T-5
A   IEFRQ.K=(INEFRQ)/STTC.K                                         T-6
L   PTGAP.K=PTGAP.J+(DT)(PTGPR.JK)                                  T-7
N   PTGAP=10                                                        T-7N
```

```
R  RTGPR.KL=(TGAP.K)(TGAPFR.K)(PRP.K+1)                                T-8
A  TGAPFR.K=MGTEF.K                                                    T-8A
      TGAP - GAP IN TECHNICAL EFFECTIVENESS
      ATE - ACTUAL TECHNICAL EFFECTIVENESS
      IEFRQ - INTRINSIC EFFECTIVENESS REQUIRED
      INEFRQ - INTRINSIC EFFORT REQUIRED (MAN-MONTHS OF EFFORT)
      STTC - SCHEDULED TIME TO COMPLETION (MONTHS)
      PTGAP - PERCEIVED GAP IN TECHNICAL EFFECTIVENESS
      RTGPR - RATE OF RECOGNITION OF TECHNICAL GAP
      TGAPFR - FRACTION OF GAP RECOGNIZED
      PRP - PRODUCTION PRESSURE
      MGTEF - MANAGEMENT EFFECTIVENESS
A  TPRES.K=TABHL(TTPRES,(PTGAP.K/TSOA.K),0,1.02,.17)                   T-9
T  TTPRES=0/.1/.15/.21/.37/.62/1.0                                     T-10
      TPRES - PRESSURE FOR IMPROVED TECHNOLOGY
      TTPRES - TABLE OF PRESSURE FOR IMPROVED TECHNOLOGY
               VERSUS PERCEIVED GAP IN TECHNICAL EFFECTIVENESS
      PTGAP - PERCEIVED GAP IN TECHNICAL EFFECTIVENESS
A  UTE.K=(ATE.K)(MGTEF.K)                                              T-11
A  MGTEF.K=MCIT.K/TSOA.K                                               T-12
      UTE - UTILIZED TECHNICAL EFFECTIVENESS
      ATE - ACTUAL TECHNICAL EFFECTIVENESS
      MGTEF - MANAGEMENT EFFECTIVENESS
      TSOA - STATE OF THE ART IN SOFTWARE DEVELOPMENT TECHNOLOGY
      MCIT - MANAGEMENT CAPABILITY TO IMPLEMENT TECHNOLOGY
L  MCIT.K=MCIT.J+(DT)(RMIMP.JK)                                        T-13
N  MCIT=TSOA                                                           T-13N
R  RMIMP.KL=(MPRES.K)(MIFF)(MCIT.K)                                    T-14
C  MIFF=.75                                                            T-14A
L  PMGAP.K=PMGAP.J+(DT)(RMGPR.JK)                                      T-15
N  PMGAP=1                                                             T-15N
R  RMGPR.KL=MAX((MGAP.K)(MGAPFR.K)(PRP.K+1),0)                         T-15
A  MGAPFR.K=MGTEF.K                                                    T-15A
A  MGAP.K=TSOA.K-MCIT.K                                                T-17
      MCIT - MANAGEMENT CAPABILITY TO IMPLEMENT TECHNOLOGY
      RMIMP - RATE OF IMPROVEMENT IN MANAGEMENT CAPABILITY
      MPRES - PRESSURE FOR IMPROVED MANAGEMENT
      MIFF - MANAGEMENT CAPABILITY IMPROVEMENT FUNDING FACTOR
      PMGAP - PERCEIVED GAP IN MANAGEMENT CAPABILTY
      RMGPR - RATE OF RECOGNITION OF MANAGEMENT CAPABILITY GAP
      MGAP - GAP IN MANAGEMENT CAPABILITY TO IMPLEMENT TECHNOLOGY
      MGAPFR - FRACTION OF MANAGEMENT GAP RECOGNIZED
      PRP - PRODUCTION PRESSURE
      MGTEF - MANAGEMENT EFFECTIVENESS
A  MPRES.K=TABHL(TMPRES,(PMGAP.K/TSOA.K),0,1.02,.17)                   T-18
T  TMPRES=0/.1/.15/.21/.37/.62/1.0                                     T-19
      MPRES - PRESSURE FOR IMPROVED MANAGEMENT
      TMPRES - TABLE OF PRESSURE FOR IMPROVED MANAGEMENT VERSUS
               PERCEIVED GAP IN MANAGEMENT CAPABILITY
```

```
*   PRODUCTION MANAGEMENT SECTOR
*   **********************************

L   EWR.K=EWR.J-(DT)(RRQADJ.JK+REPRO.JK)                          PM-1
N   EWR=(INEFRO)(1-WEER)                                          PM-1N
L   EWA.K=EWA.J+(DT)(REPRO.JK)                                    PM-2
N   EWA=0                                                         PM-2N
R   RRQADJ.KL=MAX((PRREQ.K-RCHGR)(ESCHO.K)(PRREQ.K/ABM.K),0)     PM-3
C   RCHGR=.2                                                      PM-3A
R   REPRO.KL=(EPE.K)(1+EIPR.K)                                    PM-4
N   WEER=TABHL(TWEER,MGTEF,0,1.2,.17)                            PM-5N
T   TWEER=1/.84/.67/.5/.34/.17/0                                  PM-6
      EWR - ESTIMATED WORK REMAINING (MAN-MONTHS OF EFFORT)
      EWA - ESTIMATED WORK ACCOMPLISHED (MAN-MONTHS OF EFFORT)
      RRQADJ - RATE OF ADJUSTMENTS TO REQUIREMENTS
               (MAN-MONTHS OF EFFORT PER MONTH)
      REPRO - ESTIMATED RATE OF PRODUCTION (MAN-MONTHS OF EFFORT
              PER MONTH)
      INEFRO - INTRINSIC EFFORT REQUIRED (MAN-MONTHS OF EFFORT
               PER MONTH)
      WEER - ERROR IN ESTIMATE OF WORK REQUIRED
      TWEER - TABLE OF WORK REQUIRED ESTIMATION ERROR VERSUS
              MANAGEMENT EFFECTIVENESS
      PRQADJ - PRESSURE TO RELAX REQUIREMENTS
      ESCHO - ESTIMATED SCHEDULE OVERRUN (MONTHS)
      RCHGR - RESISTANCE TO CHANGES IN REQUIREMENTS
      ABM - ADJUSTMENTS BALANCING MODIFIER
      EPE - ESTIMATED PRODUCTIVITY OF PRODUCTION ENGINEERS
            (MAN-MONTHS OF EFFORT PER MONTH)
      EIPR - ESTIMATED IMPROVEMENT TO PRODUCTION RATE
             (MAN-MONTHS OF EFFORT PER MONTH)
      MGTEF - MANAGEMENT EFFECTIVENESS
A   EPE.K=(PERAP.K)(1+PEER.K)                                     PM-7
A   PEER.K=TABHL(TPEER,MGTEF.K,0,1.2,.17)                        PM-8
T   TPEER=1/.84/.67/.50/.34/.17/0                                 PM-9
A   EIPR.K=(MXIMP.K)(PRP.K)                                      PM-10
A   MXIMP.K=(MAXPR.K/EPE.K)-1                                    PM-11
A   MAXPR.K=(PCEIT.K)(PEIT.K)+(PCEXE.K)(PEXE.K)
X   +(PCMTE.K)(PMTE.K)                                           PM-12
      EPE - ESTIMATED PRODUCTIVITY OF PRODUCTION ENGINEERS
            (MAN-MONTHS OF EFFORT PER MONTH)
      PERAP - RAW AGGREGATE PRODUCTIVITY OF PRODUCTION ENGINEERS
              (MAN-MONTHS OF EFFORT PER MONTH)
      PEER - ERROR IN ESTIMATE OF PRODUCTION ENGINEER
             PRODUCTIVITY
      TPEER - TABLE OF PRODUCTION ENGINEER PRODUCTIVITY
              ERROR VERSUS MANAGEMENT EFFECTIVENESS
      EIPR - ESTIMATED IMPROVEMENT TO PRODUCTION RATE
             (MAN-MONTHS OF EFFORT PER MONTH)
      MXIMP - MAXIMUM PRODUCTIVITY IMPROVEMENT
      MAXPR - MAXIMUM AGGREGATE PRODUCTIVITY OF PRODUCTION
      ENGINEERS (MAN-MONTHS OF EFFORT PER MONTH)
      PRP - PRODUCTION PRESSURE
```

```
A   PRP.K=TABHL(TPRP,(ESCHO.K/STTC.K),0,1.12,.17)                    PM-13
N   PRP=.5                                                           PM-13N
T   TPRP=.5/.6/.68/.82/.91/.99/1                                     PM-14
       PRP - PRODUCTION PRESSURE
       TPRP - TABLE OF PRODUCTION PRESSURE VERSUS ESTIMATED SCHEDULE
              OVERRUN AND SCHEDULED TIME TO COMPLETION
       ESCHO - ESTIMATED SCHEDULE OVERRUN (MONTHS)
       STTC - SCHEDULED TIME TO COMPLETION (MONTHS)
L   STTC.K=STTC.J+(DT)(RSCHADJ.JK-1)                                 PM-15
N   STTC=ETTC                                                        PM-15N
R   RSCHADJ.KL=MAX((PRSCH.K-RCHGS)(ESCHO.K)(PRSCH.K/ABM.K),0)       PM-16
C   RCHGS=.2                                                        PM-16A
A   ESCHO.K=MAX((ETTC.K-STTC.K),0)                                  PM-17
N   ESCHO=0                                                         PM-17N
A   ETTC.K=MAX(EWR.K/REPRO.JK,0)                                    PM-18
N   ETTC=EWR/REPRO                                                  PM-18N
       STTC - SCHEDULED TIME TO COMPLETION (MONTHS)
       RSCHADJ - RATE OF ADJUSTMENTS TO SCHEDULE
                 (MONTHS PER MONTH)
       PRSCH - PRESSURE TO RELAX SCHEDULE
       ESCHO - ESTIMATED SCHEDULE OVERRUN (MONTHS)
       RCHGS - RESISTANCE TO CHANGES IN SCHEDULE
       ABM - ADJUSTMENTS BALANCING MODIFIER
       ETTC - ESTIMATED TIME TO COMPLETION (MONTHS)
       EWR - ESTIMATED WORK REMAINING (MAN-MONTHS OF EFFORT)
       REPRO - ESTIMATED RATE OF PRODUCTION (MAN-MONTHS OF EFFORT
               PER MONTH)
A   PRSCH.K=TABHL(TPRSCH,PRP.K,0,1.12,.17)                          PM-19
T   TPRSCH=0/.05/.1/.2/.35/.75/.99                                   PM-20
A   PRREQ.K=TABHL(TPRREQ,PRP.K,0,1.12,.17)                          PM-21
T   TPRREQ=0/.05/.1/.2/.35/.75/.99                                   PM-22
       PRSCH - PRESSURE TO RELAX SCHEDULE
       TPRSCH - TABLE OF PRESSURE TO RELAX SCHEDULE VERSUS
                PRODUCTION PRESSURE
       PRREQ - PRESSURE TO RELAX REQUIREMENTS
       TPRREQ - TABLE OF PRESSURE TO RELAX REQUIREMENTS VERSUS
                PRODUCTION PRESSURE
       PRP - PRODUCTION PRESSURE
A   ABM.K=PRREQ.K+PRSCH.K                                           PM-23
       ABM - ADJUSTMENTS BALANCING MODIFIER


*  QUALITY MANAGEMENT SECTOR
*  **********************

L   DISRE.K=DISRE.J+(DT)(RDISC.JK)                                  QM-1
N   DISRE=1                                                         QM-1N
L   UDISRE.K=UDISRE.J+(DT)(REPR.JK-RDISC.JK)                       QM-2
N   UDISRE=1                                                        QM-2N
R   RDISC.KL=(TERP.K)(TQEF.K)                                       QM-3
R   RERR.KL=(1-PQEF.K)(RPRO.JK)                                     QM-4
A   TQEF.K=(SQMC.K/TSQA.K)(FQTP)                                    QM-5
C   FQTP=.5                                                         QM-5A
```

```
A   POEF.K=(SQMC.K/TSOA.K)(FQPP)                                    QM-5
C   FQPP=.5                                                         QM-5A
      DISPE - DISCOVERED REWORK IN SOFTWARE PRODUCTION
              (MAN-MONTHS OF EFFORT)
      UDISPE - UNDISCOVERED REWORK IN SOFTWARE PRODUCTION
              (MAN-MONTHS OF EFFORT)
      RDISC - RATE OF DISCOVERY OF REWORK IN PRODUCTION
              (MAN-MONTHS OF EFFORT PER MONTH)
      RERR - RATE OF ERROR IN SOFTWARE PRODUCTION
             (MAN-MONTHS OF EFFORT PER MONTH)
      TEAP - AGGREGATE PRODUCTIVITY OF TEST ENGINEERS
             MAN-MONTHS OF EFFORT PER MONTH)
      TQEF - EFFECTIVENESS OF QUALITY TESTING PROGRAM
      PQEF - EFFECTIVENESS OF QUALITY IN PRODUCTION PROGRAM
      RPRO - RATE OF PRODUCTION (MAN-MONTHS OF EFFORT PER
             MONTH)
      SQMC - SOFTWARE QUALITY MANAGEMENT CAPABILITY
      FQTP - PROPORTION OF QUALITY EFFORT DEDICATED TO QUALITY
             TESTING
      FQPP - PROPORTION OF QUALITY EFFORT DEDICATED TO QUALITY
             PRODUCTION
      MGTEF   MANAGEMENT EFFECTIVENESS
L   SQMC.K=SQMC.J+(DT)(RQMIMP.JK)                                   QM-7
N   SQMC=TSOA*2                                                     QM-7N
R   RQMIMP.KL=(QIP.K-1)(QMIFF)(SQMC.K)                             QM-8
C   QMIFF=.75                                                       QM-8A
      SQMC - SOFTWARE QUALITY MANAGEMENT CAPABILITY
      RQMIMP - RATE OF IMPROVEMENT IN SOFTWARE QUALITY MANAGEMENT
               CAPABILITY
      QIP - QUALITY IMPROVEMENT PRESSURE
      QMIFF - QUALITY MANAGEMENT IMPROVEMENT FUNDING FACTOR
A   QIP.K=TABHL(TQIP,EER.K,0,1.2,.17)                               QM-9
T   TQIP=1/1/1.1/1.4/1.8/1.99/2.0                                   QM-10
A   EER.K=MAX(((PER.K/AER)-1),0)                                    QM-11
C   AER=.1                                                          QM-11A
A   PER.K=FIFZE(0,(DISRE.K/EWA.K),EWA.K)                           QM-12
      QIP - QUALITY IMPROVEMENT PRESSURE
      TQIP - TABLE OF QUALITY IMPROVEMENT PRESSURE VERSUS
             EXCESS ERROR RATE
      EER - EXCESS ERROR RATE (MAN-MONTHS OF EFFORT PER MONTH)
      AER - ACCEPTABLE ERROR RATE (MAN-MONTHS OF EFFORT PER MONTH)
      PER - PERCEIVED ERROR RATE (MAN-MONTHS OF EFFORT PER MONTH)
      DISRE - DISCOVERED REWORK IN SOFTWARE PRODUCTION
              (MAN-MONTHS OF EFFORT)
      EWA - ESTIMATED WORK ACCOMPLISHED (MAN-MONTHS OF EFFORT)
      TQEF - EFFECTIVENESS OF QUALITY TESTING PROGRAM
```

AD-A115 555 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 9/2
WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYS--ETC(U)
MAR 82 B D MERCER

UNCLASSIFIED AFIT/GCS/MA/82M-3 NL

Vita

Bradford Douglas Mercer was born 30 May 1951 at Hamilton AFB near San Rafael, California. He graduated from high school in San Antonio, Texas in 1969 and attended Texas A&M University in College Station and San Antonio College in San Antonio, Texas, receiving the degree of Bachelor of Arts with major in economics from the University of Texas at Austin in May 1974. He was a distinguished graduate of Air Force ROTC at the University of Texas and was commissioned upon graduation in 1974. He entered active duty and began navigator training at Mather AFB, California in July 1974. Receiving his navigator wings in March 1975, he entered electronic warfare officer training, also at Mather AFB, graduating in October 1975. He served as a B-52G electronic warfare officer, instructor and flight examiner with the 320th Bombardment Wing before assuming duties as the Wing Mission Development Officer in April 1979. He completed the requirements for a Master of Science in Systems Management degree through the University Southern California (USC) prior to entering the Air Force Institute of Technology (AFIT) in August 1980, and was awarded that degree by USC in September 1980. He will receive the degree of Master of Science in Computer Systems from AFIT

in March 1982. He is a member of the Institute of Electrical and Electronic Engineers and the Air Force Association.

Permanent Address: 4791 Oak Twig Way
Carmichael, California 95608

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GCS/MA/82M-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and subtitle) WEAPON SYSTEM SOFTWARE ACQUISITION AND SUPPORT: A THEORY OF SYSTEM STRUCTURE AND BEHAVIOR		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bradford D. Mercer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 1982 13. NUMBER OF PAGES 183
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

15 APR 1982

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

~~FREDRIC C. LYNCH, Major, USAF~~
~~Director of Public Affairs~~

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

POLICY ANALYSIS
POLICY MODELLING
SYSTEM DYNAMICS
SOFTWARE ACQUISITION AND SUPPORT
SOFTWARE MANAGEMENT
SOFTWARE ENGINEERING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The system for acquiring and supporting weapon system software was investigated through the methodology of system dynamics, a technique for studying the structure of socio-technical systems and how that structure determines their behavior. Conceptual and mathematical models of a generalized software production process and the influences upon that process were developed. The

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

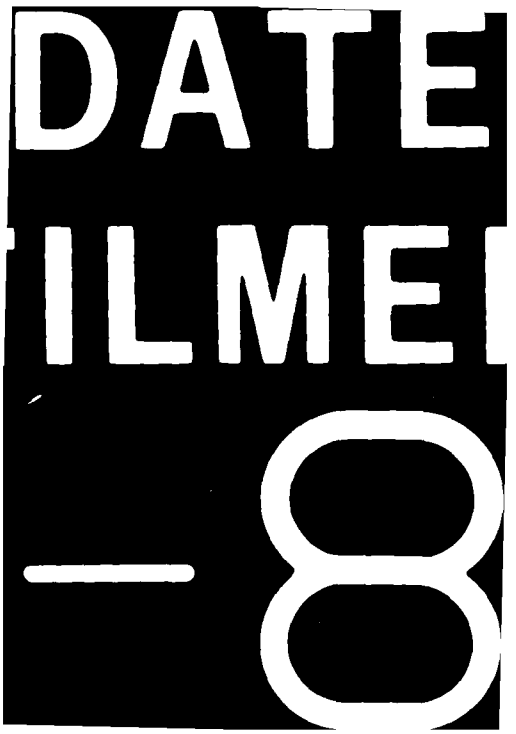
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

mathematical model was translated into a continuous simulation computer model using the DYNAMO language.

These models can be examined by managers at all levels in the acquisition and support process in order to increase their understanding of the complexities and interactions of system decisional structures. Such understanding is foundational to the analysis and formulation of policy to guide and improve the performance of that management system.

Discussion of experimental and nonexperimental modes of research lead to recommendations for future experimental research for policy analysis, formulation, and implementation.


DATE
ILME
8